



UNIVERSIDAD CARLOS III DE MADRID

INGENIERÍA INFORMÁTICA

TRABAJO FIN DE CARRERA

GENERADOR DE IMÁGENES PANORÁMICAS

AUTOR:	FRANCISCO JAVIER GARCÍA VILLASUR
TUTOR:	FELIX GARCÍA CARBALLEIRA
TUTOR:	RAQUEL CEDAZO LEÓN

FEBRERO 2010

Copyright © 2010 Francisco Javier GARCÍA VILLASUR

Está permitido copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes, ni texto de Cubierta Frontal, ni Cubierta Posterior.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the FreeSoftware Foundation; with no Invariant Sections, no Front-Cover Text, and no Back-Cover Texts. A copy of the licence is included in the section entitled “GNU Free Documentation License”.

Agradecimientos

Este trabajo va dedicado a todas las personas que me han estado animando todos estos años para que intentara terminar el trabajo fin de carrera

Índice

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	1
1.3	Trabajo del autor.....	2
1.4	Estructura de la memoria.....	2
2	Estado del arte.....	5
2.1	Mosaicos de imágenes.....	5
2.1.1	Astrofotografía.....	5
2.1.2	¿Que es un mosaico?.....	6
2.1.3	Generación de mosaicos.....	6
2.1.3.1	Toma de imágenes.....	7
2.1.3.2	Alineamiento de imágenes.....	7
2.1.3.2.1	SIFT.....	8
2.1.3.2.2	SURF.....	9
2.1.3.3	Proyección de imágenes.....	9
2.1.3.4	Fusión de imágenes.....	10
2.1.3.4.1	Calibración de Imágenes.....	10
2.1.3.4.2	Transformación de imágenes.....	12
2.1.3.4.3	Mezcla de imágenes.....	13
2.1.3.4.3.1	Corrección del color.....	13
2.1.3.4.3.2	HDR.....	13
2.1.3.4.3.3	Compensación de movimiento.....	13
2.2	Herramientas mosaicos.....	13
2.2.1	Comerciales.....	14
2.2.1.1	AutoStitch.....	14
2.2.1.2	Autopano Pro.....	14
2.2.1.3	PTGui.....	15
2.2.1.4	PanoWeaver.....	16
2.2.1.5	Otros.....	17
2.2.2	Código abierto.....	17
2.2.2.1	Hugin.....	18
2.2.2.2	PanoPoints.....	18
2.2.2.3	Panorama Tools Open GUI.....	19
2.2.2.4	iMerge.....	20
2.2.2.5	AstroMosaic.....	20
2.2.2.6	Otros.....	21
2.2.3	Conclusiones.....	21
2.3	Proyecto Ciclope.....	23
2.3.1	Módulos.....	23
2.4	Aportaciones del TFC.....	24
3	Descripción del trabajo.....	26
3.1	Planificación.....	26
3.2	Primeros pasos.....	26

3.2.1 Premisas de partida.....	26
3.2.2 Fotografía digital.....	28
3.2.3 Elección del producto.....	28
3.2.3.1 Características.....	30
3.2.3.2 Ventajas.....	32
3.2.3.3 Funcionamiento.....	32
3.2.3.4 Suite Hugin.....	33
3.2.3.4.1 nona.....	33
3.2.3.4.2 enblend.....	33
3.2.3.4.3 Enfuse.....	33
3.2.3.4.4 Fulla.....	34
3.3 Análisis de la integración.....	35
3.3.1 Análisis de las librerías de composición.....	35
3.3.2 Análisis de la interfaz de usuario.....	36
3.3.2.1 Primeros pasos.....	38
3.3.2.2 Generación del prototipo.....	38
3.3.2.3 Capacidades multiidioma.....	41
3.3.3 Integración cliente servidor.....	41
3.3.3.1 Llamadas a procedimiento remoto.....	42
3.3.3.2 Arquitectura servidor.....	42
3.3.3.3 Lógica de negocio y Acceso a Datos.....	45
3.3.4 Visor de imágenes.....	46
3.3.4.1 IIP.....	47
3.3.4.2 Formato Piramidal.....	47
4 Diseño de la aplicación.....	49
4.1 Requisitos de usuario.....	49
4.1.1 Requerimientos Formales.....	50
4.1.1.1 Requerimientos funcionales.....	50
4.1.1.1.1 Gestión Usuarios.....	51
4.1.1.1.2 Gestión Álbumes.....	51
4.1.1.1.3 Gestión Etiquetas.....	54
4.1.1.1.4 Gestión Imágenes.....	55
4.1.1.1.5 Gestión Mosaicos.....	56
4.1.1.2 Requerimientos no-funcionales.....	59
4.1.1.2.1 Desarrollo.....	60
4.1.1.2.2 Tecnológicos.....	61
4.2 Modelo de casos de uso.....	61
4.2.1 Diagrama de casos de uso.....	61
4.2.2 Descripción de los actores.....	61
4.2.3 Casos de uso en formato breve.....	63
4.2.3.1 Gestión Usuarios.....	63
4.2.3.2 Gestión Álbumes.....	64
4.2.3.3 Gestión Etiquetas.....	70
4.2.3.4 Gestión Imágenes.....	73
4.2.3.5 Gestión Mosaicos.....	76
4.3 Matriz de Trazabilidad.....	80
4.3.1 Gestión de usuarios.....	80
4.3.2 Gestión de álbumes.....	81

4.3.3 Gestión de etiquetas.....	81
4.3.4 Gestión de imágenes.....	82
4.4 Diseño de la base de datos.....	82
4.4.1 álbumes.....	84
4.4.2 config.....	84
4.4.3 imagenes.....	85
4.4.4 label_album.....	85
4.4.5 labels.....	86
4.4.6 lens.....	86
4.4.7 users.....	86
5 Tecnologías utilizadas.....	88
5.1 Tecnologías de programación.....	88
5.1.1 Cliente.....	88
5.1.2 Servidor.....	90
5.1.2.1 JNI.....	91
5.1.2.2 Spring.....	91
5.1.2.3 Ibatis.....	91
5.2 Tecnologías de desarrollo.....	92
5.2.1 Eclipse IDE SDK.....	92
5.3 Tecnologías de despliegue.....	93
5.3.1 Maven.....	93
5.4 Tecnologías de ejecución.....	94
5.4.1 Tomcat.....	94
5.4.2 Apache.....	94
5.4.3 Mysql.....	95
6 Manual de usuario.....	96
6.1 Componentes.....	96
6.2 Estructura de la aplicación.....	97
6.2.1 Pantalla Autenticación.....	97
6.2.2 Pantalla principal.....	98
6.2.2.1 Pestaña “Acerca de”.....	99
6.2.2.2 Pestaña “Asistente”.....	99
6.2.2.2.1 Panel Superior.....	100
6.2.2.2.2 Panel Izquierdo.....	102
6.2.2.2.3 Panel Derecho.....	110
6.2.2.2.4 Panel Central.....	113
6.2.2.2.5 Panel Inferior.....	115
6.2.3 Pestaña “Información Cámara”.....	117
6.2.3.1 Panel Izquierdo.....	117
6.2.3.2 Panel Central.....	121
6.2.3.3 Panel Inferior.....	123
6.2.4 Pestaña “Previsualización Panorama”.....	124
6.2.4.1 Pestaña Información.....	125
6.3 Proceso de generación de mosaicos.....	127
6.3.1 PASO1. Preparación del álbum contenedor.....	127
6.3.2 PASO2. Carga de imágenes.....	132
6.3.3 PASO3. Subida de imágenes para composición.....	135
6.3.4 PASO4. Alineación de imágenes.....	140

6.3.5 PASO5. Generación de la panorámica.....	146
6.3.6 PASO6. Resultado Final.....	149
7 Manual de instalación.....	152
7.1 Manual de compilación.....	154
7.1.1 Requisitos.....	154
7.1.2 Compilación librerías mosaicos.....	155
7.1.2.1 Hugin.....	156
7.1.2.1.1 Entorno inicial.....	156
7.1.2.1.2 Enblend.....	156
7.1.2.1.3 libpano13.....	157
7.1.2.1.4 autopano-sift-c.....	158
7.1.2.1.5 exifttool.....	158
7.1.2.1.6 Hugin.....	159
7.1.2.1.7 PanoGLView.....	160
7.1.2.2 Boost-Process.....	161
7.1.2.3 LibMosaic.....	162
7.1.3 Compilación módulo cliente/servidor GWT.....	165
7.1.3.1 Requisitos.....	165
7.1.3.2 Compilación Módulos.....	165
7.1.3.2.1 ImageSlideShow.....	165
7.1.3.2.2 MosaicModel.....	165
7.1.3.2.3 MosaicImageComposer-Server.....	166
7.1.3.2.4 MosaicImageComposer-Client.....	166
7.1.3.3 Generación del WAR (Web Archive).....	167
7.2 Manual de instalación.....	168
7.2.1 Instalación de servicios.....	168
7.2.1.0.1 Requisitos.....	169
7.2.1.1 Instalación JDK.....	169
7.2.1.2 Instalación Apache Tomcat.....	169
7.2.1.2.1 Preparación.....	170
7.2.1.2.2 Personalización.....	170
7.2.1.2.3 Autoarranque.....	171
7.2.1.2.4 Datos Configuración.....	172
7.2.1.2.4.1 Arranque.....	173
7.2.1.2.4.2 Parada.....	173
7.2.1.3 Instalación Mysql.....	174
7.2.1.3.1 Preparación.....	174
7.2.1.3.2 Autoarranque.....	175
7.2.1.3.3 Datos Configuración.....	175
7.2.1.3.3.1 Arranque.....	176
7.2.1.3.3.2 Parada.....	176
7.2.1.4 Instalación Apache Httpd.....	177
7.2.1.4.1 Preparación.....	177
7.2.1.4.2 Personalización.....	177
7.2.1.4.3 Autoarranque.....	178
7.2.1.4.4 Datos Configuración.....	179
7.2.1.4.4.1 Arranque.....	179
7.2.1.4.4.2 Parada.....	179

7.2.2 Instalación de aplicativos.....	180
7.2.2.1 Plugins Apache.....	180
7.2.2.1.1 Fast-Cgi.....	180
7.2.2.1.1.1 Preparación.....	180
7.2.2.1.2 Mod-Proxy.....	181
7.2.2.1.2.1 Preparación.....	181
7.2.2.1.3 Iipmooviewer.....	182
7.2.2.1.4 Iipmooviewer-server.....	183
7.2.2.1.4.1 Requisitos.....	183
7.2.2.1.4.2 Preparación.....	183
7.2.2.1.5 Iipmooviewer-client.....	185
7.2.2.1.5.1 Preparación.....	185
7.2.2.2 Software Mosaicos.....	186
7.2.2.2.1 Requisitos.....	186
7.2.2.2.2 Instalación Productos.....	186
7.2.2.2.2.1 MosaicImageComposer.....	186
7.2.2.2.2.2 libMosaic.....	187
7.2.2.2.2.3 libProducts.....	187
7.2.2.2.3 Configuración.....	188
7.2.2.2.4 Carga de datos.....	190
7.2.3 Carga de Albums.....	191
7.2.4 Test de aplicativos.....	191
8 Conclusiones y líneas futuras.....	192
8.1 Conclusiones.....	192
8.2 Líneas futuras.....	192
Bibliografía.....	195
9 Apéndice 1. Presupuesto.....	196
9.1 Fases.....	196
9.2 Recursos.....	196
9.3 Recursos tecnológicos.....	197
9.4 Presupuesto de Costes.....	197
10 Apéndice 2. Fotografía digital.....	198
10.1 Conceptos Básicos.....	198
10.1.1 Distancia Focal.....	198
10.1.2 Diafragma y abertura.....	198
10.1.3 Obturador.....	198
11 Apéndice 3. Acrónimos.....	200

Índice de figuras

Figura 1: Distorsión negativa (angular)	10
Figura 2: Sin distorsión	10
Figura 3: Distorsión positiva (teleobjetivo)	10
Figura 4: Corrección de perspectiva vertical	11
Figura 5: Efecto Barril	11
Figura 6: Efecto vignetting	12
Figura 7: Efecto Aberración cromática (Azul)	12
Figura 8: AutoStich ScreenShot	15
Figura 9: PTGui ScreenShot	16
Figura 10: PTGui	17
Figura 11: EasyPano ScreenShot	18
Figura 12: Hugin ScreenShot	19
Figura 13: PanoPoints ScreenShot	20
Figura 14: Panorama Tools Open Gui ScreenShot	21
Figura 15: Estructura proyecto Cíclope	25
Figura 16: Planificación	27
Figura 17: Hugin1. Inclusión de puntos	30
Figura 18: Hugin2	30
Figura 19: Arquitectura Hugin	32
Figura 20: Arquitectura del aplicativo	35
Figura 21: Porcentaje Uso Navegadores. Septiembre 2009	37
Figura 22: Arquitectura Componentes GWT	39
Figura 23: Boceto pantallas	40
Figura 24: Panel SlideShow con soporte Drag & Drop	41
Figura 25: Arquitectura RPC en GWT	42
Figura 26: Matriz de Trazabilidad. Gestión de Usuarios	78
Figura 27: Matriz de Trazabilidad. Gestión de Álbumes	78
Figura 28: Matriz de Trazabilidad. Gestión de Etiquetas	79
Figura 29: Matriz de Trazabilidad. Gestión de Imágenes	80
Figura 30: Arquitectura Proyecto	86
Figura 31: Pantalla Login	95
Figura 32: Panel de Pestañas	95
Figura 33: Pestaña Acerca De	96
Figura 34: Pestaña Asistente	97
Figura 35: Panel Menú	98
Figura 36: Pestaña Asistente Consola	98
Figura 37: Pestaña Asistente Config	99
Figura 38: Listado de Albums	100
Figura 39: Listado de Etiquetas	100
Figura 40: Árbol de Albums	101
Figura 41: Nuevo Álbum	102
Figura 42: Selección de un Álbum	102
Figura 43: Subida de imágenes Paso1	103
Figura 44: Subida de imágenes Paso2	104
Figura 45: Añadir Etiqueta Paso1	105

Figura 46: Añadir Etiqueta Paso2	105
Figura 47: Creación de nueva etiqueta	106
Figura 48: Panel Derecho	107
Figura 49: Información de imagen	108
Figura 50: Información de imagen	109
Figura 51: Álbum con panorámica	111
Figura 52: Álbum sin panorámica	111
Figura 53: Panel central	112
Figura 54: Panel inferior	113
Figura 55: Drag & Drop	114
Figura 56: Pestaña Camera	115
Figura 57: Panel de Información Geométrica	116
Figura 58: Lentes disponibles	117
Figura 59: Aplicación de lentes en imágenes	118
Figura 60: Lista de imágenes	118
Figura 61: Grupos no conectados	119
Figura 62: Agrupación de imágenes	119
Figura 63: Previsualización de imágenes	120
Figura 64: Previsualización de imágenes. Desplazamiento	120
Figura 65: Pestaña Camera	121
Figura 66: Pestañas Previsualización	121
Figura 67: Información alineación	122
Figura 68: PASO 1.1. Creación del álbum 1	123
Figura 69: PASO 1.1. Creación del álbum 2	123
Figura 70: PASO 1.2. Modificación del álbum 1.	124
Figura 71: PASO 1.2. Modificación del álbum 2.	124
Figura 72: PASO 1.3. Creación de categoría 1	125
Figura 73: PASO 1.3. Creación de categoría 2	125
Figura 74: PASO 1.4. Modificación de categoría 1	126
Figura 75: PASO 1.4. Modificación de categoría 2	126
Figura 76: PASO5. Asignación de categoría 1	127
Figura 77: PASO5. Asignación de categoría 2	127
Figura 78: PASO 1.5. Asignación de categoría3	128
Figura 79: PASO 1.5. Asignación de categoría 4	128
Figura 80: PASO 2.1. Carga de Imágenes 1	129
Figura 81: PASO 2.1. Carga de Imágenes 2	129
Figura 82: PASO 2.1. Carga de imágenes 3	130
Figura 83: PASO 2.1. Carga de imágenes 4	130
Figura 84: PASO 2.1. Carga de imágenes 5	130
Figura 85: PASO 2.2. Verificación de Imágenes 1	131
Figura 86: PASO 3.1 Subida de Imágenes para validación 1	132
Figura 87: PASO 3.1 Subida de Imágenes para validación 2	132
Figura 88: PASO 3.2 Verificación de imágenes 1	133
Figura 89: PASO 3.2 Verificación de imágenes 2	134
Figura 90: PASO 3.2 Verificación de imágenes 3	135
Figura 91: PASO 3.2 Verificación de imágenes 4	135
Figura 92: PASO 3.2 Verificación de imágenes 5	136
Figura 93: PASO 4.1. Alineación Imágenes 1	137

Figura 94: PASO 4.2. Arreglo Imágenes 2	138
Figura 95: PASO 4.2. Arreglo Imágenes 2	138
Figura 96: PASO 4.2 Arreglo Imagen 3	138
Figura 97: PASO 4.2 Arreglo Imagen 4	139
Figura 98: PASO 4.3 Previsualización Imagen 1	140
Figura 99: PASO 4.3 Previsualización Imagen 2	141
Figura 100: PASO 4.3 Previsualización Imagen 3	141
Figura 101: PASO 5.1. Generación Panorámica 1	142
Figura 102: PASO 5.1. Generación Panorámica 2	143
Figura 103: PASO 5.2 Previsualización Imagen 1	143
Figura 104: PASO 5.2 Previsualización Imagen 2	144
Figura 105: PASO 5.1. Previsualización Panorama 1	144
Figura 106: PASO 6.1. Resultado Final 1	145

1 Introducción

1.1 Motivación

Este proyecto se enmarca dentro de un proyecto más amplio denominado Ciclope. El objeto es permitir que los usuarios dispongan de la posibilidad de generar mosaicos con las imágenes tomadas por el telescopio del proyecto Ciclope Astro a través de una aplicación web.

Además de este objetivo, conocer y aplicar las técnicas de fotografía digital para poder estudiar y aplicar los valores que afectan a la composición de las imágenes se ha valorado como una motivación adicional para la realización de este trabajo.

Por último y no menos importante, el conocer una nueva técnica como Google Web Toolkit para el desarrollo de aplicaciones Web de una forma sencilla y sin tener que preocuparse de complicadas librerías y componentes diversos ha sido un factor importante en la elección de este trabajo.

1.2 Objetivos

El objetivo principal de este trabajo es el análisis, diseño y desarrollo de una aplicación web que permita a los usuarios la generación y posterior visualización de mosaicos de imágenes.

Esta aplicación está integrada como un módulo adicional dentro del proyecto Ciclope y por tanto va a permitir que los usuarios registrados en la plataforma pueden hacer uso de la misma. En el proyecto Ciclope el laboratorio virtual que forma parte del núcleo del proyecto permite que los usuarios puedan tomar instantáneas fotográficas del espacio (astrofotografías). Estas instantáneas son almacenadas dentro del sistema y permite que posteriormente el usuario puede analizarlas más detalladamente descargándoselas a su ordenador.

Las instantáneas que se toman del espacio en muchas ocasiones no pueden abarcar en su totalidad el objeto deseado por lo que la necesidad de tomar varias fotografías del objeto se hace evidente y necesaria.

Es por tanto muy importante disponer en entornos como éste, de técnicas que permitan que las imágenes que se hayan generado se puedan componer en una imagen completa fusionando todos los fragmentos tomados por el usuario. Esta técnica se conoce como composición de mosaicos.

Se pretende por tanto que estos usuarios puedan, sin necesidad de descargarse las imágenes en su ordenador y utilizar complejos programas informáticos especializados, generar una imagen que agrupe todas las imágenes generadas y que posteriormente puedan visualizar o descargar si lo desean para un mejor estudio.

Se desarrollarán por tanto los módulos necesarios para proporcionar las siguientes funcionalidades:

- Gestión de contenidos, que englobará todas las funcionalidades necesarias para que los usuarios puedan subir contenidos al sistema para su posterior proceso.
- Gestión de mosaicos, que englobará todas las funcionalidades necesarias para que los usuarios puedan generar y visualizar mosaicos de imágenes con la información almacenada en el sistema.

1.3 Trabajo del autor

El autor ha centrado su trabajo en un primer lugar en el estudio de diversas alternativas que permitieran disponer de un software abierto con el que realizar estas operaciones de generación de mosaicos.

Los pasos iniciales han sido analizar diversas aplicaciones disponibles en el mercado tanto de código abierto como herramientas de pago con el objetivo de tener una idea clara de las distintas posibilidades existentes así como de los resultados de cada una de las aplicaciones.

Un punto importante con el que se ha trabajado y que ha condicionado la búsqueda de las aplicaciones es el requisito fundamental consistente en ser una módulo web accesible desde una navegador . La mayoría de las aplicaciones que se han evaluado son aplicaciones de escritorio ya que el proceso de generación de mosaicos es un proceso costoso a nivel computacional y que puede exigir en algunos casos adaptaciones muy precisas casi a nivel manual para conseguir unos resultados aceptables.

De las aplicaciones evaluadas, de las que se realizará un estudio en el capítulo 2.2, se ha buscado como característica importante la modularidad de las mismas y la existencia en la medida de lo posible de API's para poder integrar sus características en un motor Web. La modularidad permite que las aplicaciones se puedan adaptar de una forma sencilla a nuevos algoritmos que surjan en el mercado y que ofrezcan mejores resultados mientras que el API permite una integración sencilla en un tercero.

Como último punto se ha integrado dicha aplicación dentro de Ciclope Astro con el fin de disponer de un mecanismo estándar de acceso al sistema para que los usuarios registrados en el sistema pudieran hacer uso de la aplicación.

1.4 Estructura de la memoria

El presente trabajo ha sido estructurado en una serie de capítulos principales así como varios apéndices que se han considerado de interés y utilidad.

A continuación se explica cada uno de ellos:

Este **primer capítulo**, “Introducción”, realiza una breve presentación del proyecto, describiendo su finalidad y el trabajo realizado.

Capítulo 2: “Estado del Arte”, hace un estudio de sistemas similares dedicados a la generación de mosaicos.

Capítulo 3: “Descripción del Trabajo”, donde se hace una descripción del proyecto y las fases del mismo.

Capítulo 4: “Diseño de la aplicación”, describe los casos de uso, el modelo de la base de

datos y la estructura de archivos.

Capítulo 5: “Tecnologías utilizadas”, pretende explicar cada una de las tecnologías que se han usado para el desarrollo del proyecto.

Capítulo 6: “Manual de usuario”, explica cómo utilizar la aplicación desde los diferentes perfiles.

Capítulo 7: “Manual de instalación”, explica cómo instalar todos los productos utilizados y la configuración necesaria para un correcto funcionamiento.

Capítulo 8: “Conclusiones y líneas futuras”, explica las conclusiones que se han obtenido al desarrollar el trabajo y cual puede ser el roadmap del producto.

Apendice1: “Presupuesto”, donde se indica el presupuesto del proyecto.

Apendice2: “Fotografía digital”, donde se da un repaso de términos utilizados en la fotografía digital y que puede ser interesante conocer para tener un conocimiento más extenso sobre el trabajo realizado.

2 Estado del arte

Es una de las primeras etapas que debe realizarse dentro de un proyecto de investigación, puesto que su elaboración, que consiste en realizar un estudio detallado del tema que se pretende investigar, permite determinar cómo ha sido tratado el tema, cómo se encuentra en el momento de realizar la propuesta de investigación y cuáles son las tendencias al respecto. De la misma forma permite elegir de las tecnologías disponibles aquella que se puede adaptar mejor a la realización del trabajo.

Para su elaboración, es recomendable establecer un período de tiempo, de acuerdo con los objetivos de la investigación, tras el cual se deberá decidir qué solución de las estudiadas es la más adecuada.

En este capítulo se realiza un estudio teórico de las tecnologías disponibles para la generación de mosaicos así como una descripción de distintos programas que hacen uso de estas tecnologías. En el último apartado se hace una descripción del proyecto Ciclope, en el cual se engloba el presente trabajo el cual se desplegará como un nuevo módulo dentro del proyecto global.

2.1 Mosaicos de imágenes

2.1.1 Astrofotografía

Las fotografías del espacio, por ejemplo las lunares, presentan unas características idóneas para generar mosaicos. Si se desea fotografiar la totalidad o gran parte de la Luna, se deben usar exposiciones directas, o usar lentes grandes para conseguir pocos aumentos con el objetivo de no entrar directamente en la superficie del cuerpo celeste (dependiendo de la distancia focal del telescopio). Las fotografías que se obtienen, son excelentes para apreciar todas las formas, figuras, manchas y comparar sus tamaños en toda la superficie lunar, pero con falta de detalles.

Para poder entrar en más detalle es preciso utilizar lentes más pequeñas para hacer más grande el cuerpo enfocado. El resultado será una astrofotografía, quizás muy rica en formas y detalles, pero únicamente de una porción del cuerpo. Por más que se intenten intercambiar lentes para conseguir diferentes aumentos, siempre se obtiene uno u otro resultado, amplias zonas con pocos detalles o viceversa, siempre faltará una parte de lo que se desea. En este momento donde las capacidades de las herramientas de generación de mosaicos entran en escena.

La fotografía de cuerpos celestes presenta una serie de problemas distintos que la fotografía tradicional, por ejemplo el nivel de visibilidad con la que los objetos celestes pueden ser observados. Los factores atmosféricos que contribuyen al empeoramiento de la visibilidad son, entre otros: las nubes, la neblina, la contaminación luminosa, el viento... Incluso la Luna es un factor determinante de empeoramiento cuando se pretende observar objetos del cielo profundo. Tan importante es el término que las prestaciones de un telescopio dependen en gran medida de estas condiciones atmosféricas.

Los objetos de luminosidad débil o difusa necesitarán más tiempo que los más luminosos o puntuales. Si estos objetos fueran estáticos no habría mayor problema pero, debido a su desplazamiento a través de la bóveda celeste, en ocasiones se deberá modificar continuamente el encuadre para conseguir un tiempo de exposición suficiente para obtener la imagen.

2.1.2 ¿Que es un mosaico?

En los últimos años los avances en la fotografía digital y la reducción en precios de las cámaras digitales ha posibilitado que los usuarios puedan explorar con diversas técnicas para generar y modificar imágenes permitiéndoles realizar personalizaciones de las instantáneas utilizando software muy avanzado.

Una de las características muy de moda en el mundo de la fotografía digital son las **fotos panorámicas**.

Generar un **mosaico o panorámica** consiste en, a partir de una serie de fotografías de alta definición, tratadas de forma individual corrigiéndose aspectos tales como la distorsión, los reflejos, el equilibrio de color, enfoque, etc., generar una única imagen que sea la unión de las fotos individuales.

Existen multitud de técnicas y variantes para obtener instantáneas en formato panorámico, algunas se basan en cámaras diseñadas específicamente para estos formatos especiales, otras se consiguen mediante el uso de complejos programas de ordenador, y otras utilizando ambas técnicas. El fin de todas ellas es obtener una fotografía que muestre, de un modo u otro, un encuadre que abarque un ángulo mucho mayor que el obtenido con las cámaras y los objetivos convencionales.

El problema de las cámaras convencionales para la toma de panorámicas es el tamaño del CCD (Charge Coupled Device, sistema de carga acoplada). El CCD de una cámara es básicamente el sensor de la misma y es el responsable de captar la luz que ingresa a la cámara. Su tamaño en la mayoría de las cámaras es muy pequeño y, por tanto, sólo puede captar un porcentaje muy pequeño del horizonte.

El número de posibilidades que tenemos para generar un mosaicos es bastante amplio y su aplicación va desde la fotografía casera hasta la fotografía profesional. Con un mosaico se puede, por ejemplo, generar una sola fotografía panorámica de hasta un ángulo de 360°. Estas fotos panorámicas de 360° se pueden utilizar para la generación de una técnica muy de moda con Internet como son las **visitas virtuales** para la realización de *tours* en determinados edificios .

En el apéndice Apéndice 2. Fotografía digital se realiza un repaso rápido de términos de fotografía digital que pueden ser útiles para un mejor entendimiento del trabajo realizado.

2.1.3 Generación de mosaicos

La inmensa mayoría de los programas analizados en este estudio siguen una serie de fases a la hora de generar las panorámicas:

2.1.3.1 Toma de imágenes

A la hora de tomar las distintas panorámica existen una serie de técnicas y recomendaciones que es preciso tomar en cuenta para que el resultado sea el esperado.

- Utilizar en la mayoría de los casos un **trípode** con el objetivo de que la posición de las distintas tomas sea siempre la misma. En el caso de no existir trípode intentar en la medida de lo posible mantener la misma posición horizontal.
- Ajustar la velocidad de **obturación** y la **apertura**. El gran problema que se puede plantear a la hora de hacer fotos panorámicas es la variación de la luz que se produce entre unas tomas y otras. Si la cámara está en modo automático (normalmente es el de por defecto en las cámaras compactas), es probable que la cámara varíe las condiciones de velocidad de obturación y apertura del diafragma entre foto y foto.
- Fija el balance de **blancos**. Por defecto, las cámaras digitales suelen tener el balance de blancos fijado en un modo automático. Dependiendo de las condiciones de iluminación que haya en la escena, puede ser que la cámara realice distintas mediciones y consecuentes cambios de color como consecuencia de la medición de la temperatura del color.
- Ajustar la **distancia focal**. Si se utilizan grandes angulares, puede que la distancia focal elegida provoque algún tipo de distorsión en los extremos. Utilizando focales algo más largas se suele reducir el problema de distorsión.
- Dejar espacio suficiente para solapar las fotos. A la hora de montar unas fotos sobre otras, los programas para hacer panorámicas buscan puntos coincidentes entre foto y foto (sift, surf, etc.) para solapar las fotos y poder crear la foto panorámica con garantías.

2.1.3.2 Alineamiento de imágenes

Las técnicas de alineamiento tienen como objetivo encontrar una serie de características en las imágenes que permitan ser usadas para comparar características comunes entre ellas.

El método más utilizado es el que tiene como objeto encontrar lo que se denomina **descriptor**. Los descriptores son el primer paso para poder encontrar la conexión entre los píxeles contenidos en una imagen digital y aquellas características que podemos observar visualmente en una imagen. Un descriptor local es un vector de características que es calculado sobre una pequeña región de interés de la imagen.

Para obtener estos descriptores se necesita disponer de un software que sea capaz de analizar objetos dentro de una imagen y un algoritmo que describa dicho objeto de forma genérica.

El algoritmo utilizado funcionará sobre una imagen con objetos sobre fondo negro o blanco y realizará un barrido de dicha imagen hasta encontrar todos los puntos que pertenecen al objeto. Se usará un vector que será el descriptor de objeto y donde se almacenará valores en función de la ubicación de los puntos que pertenecen al objeto en cuestión.

2.1.3.2.1 SIFT

El algoritmo más comúnmente utilizado surgió en 1999 de la mano de David Lowe el cual propuso una técnica de descripción de objetos invariante a escala y rotaciones (Scale-Invariant Feature Transform [SIFT]).

La técnica se dividía en 2 partes diferenciadas:

1. **El detector SIFT** que utilizaba la combinación de varias técnicas para detectar objetos:
 - **La representación espacio-escala de una imagen**: Esta técnica consiste en crear una familia de imágenes suavizadas a diferentes niveles de detalle (definido por un parámetro de escala t). El detector SIFT creaba el conjunto de imágenes en un rango entre t y $2t$ llamado octava.
 - **Pirámides gaussianas**: Formadas por la familia de imágenes suavizadas donde la base contiene la más alta resolución. En el caso del detector SIFT, a partir de la escala $2t$ se divide a la mitad de resolución y se usa esta resolución para la siguiente octava.
 - **Filtro Log**: Teniendo en cuenta que el espacio escala representaría la familia de imágenes y la pirámide gaussiana es el núcleo de suavizado de la imagen, suponemos que los cruces por cero de estas ecuaciones representarían los bordes del objeto, y los extremos permiten detectar manchas.
 - En el caso del detector SIFT sustraen dos escalas consecutivas. Se compara cada píxel con todos sus vecinos y sólo se seleccionan los extremos de espacio-escala que sean mayores o menores que sus vecinos.
2. **Descriptor SIFT** basado en Histogramas de orientación del gradiente: Este descriptor es invariante a rotación ya que se realiza una normalización en rotación que permite comparar puntos en varias orientaciones, donde la orientación de un punto es la orientación dominante del gradiente en su vecindad.

El gradiente es un vector bidimensional cuyos componentes están dados por las primeras derivadas de las direcciones verticales y horizontales. Podemos definir el gradiente para cada punto de la imagen, como el vector que apunta en dirección del incremento máximo posible de intensidad, y la magnitud del gradiente del vector corresponde a la cantidad de cambio de intensidad en esa dirección. El gradiente nos proporciona una idea del cambio en un punto.

Los pasos realizados para obtener el descriptor **SIFT** son los siguientes:

- Calcular la magnitud y la orientación del gradiente en la vecindad del punto utilizando la primera imagen suavizada.
- Una vez tenemos la orientación de todos sus vecinos, creamos un histograma, donde su máximo determinará la orientación del punto. De aquí podremos obtener los ejes locales para cada punto de interés.
- En el caso de existir en el histograma picos superiores al 80% del máximo se genera un nuevo punto para cada pico con su orientación correspondiente.
- Se segmenta la vecindad en regiones de 4×4 píxeles.
- Se genera un histograma de orientación de gradiente para cada región usando una

ponderación Gaussiana de ancho 4 píxeles.

- Hay que tener en cuenta que un pequeño desplazamiento espacial provoca que la contribución de un píxel pase de un segmento a otro y cambie la descripción. En el caso de una pequeña rotación la contribución podría pasar de una orientación a otra.
- Para evitar este problema un píxel contribuye a todos sus vecinos, multiplicando la contribución por un peso $1-d$, donde d es la distancia al centro del segmento.
- Por último se obtiene un histograma tridimensional de $4 \times 4 \times 8 = 128$ casillas considerando 8 direcciones principales y formando un vector con el valor de todas las casillas. Este vector se genera para cada punto de interés.

El descriptor SIFT creado por David Lowe es parcialmente robusto a cambios de iluminación, puntos de vista, se calcula rápido y es muy distintivo; pero también tiene inconvenientes ya que no es robusto a deformaciones rígidas o elásticas de objetos. Esto es debido a que a la hora de describir se basa en la creación del vector descriptor a partir de las normales de los puntos de interés del objeto y eso provocaría que pequeños cambios de curvatura varíen radicalmente el descriptor [SIFT].

2.1.3.2.2 SURF

Otro descriptor creado posteriormente al de David Lowe en 1999 fue el denominado [SURF]. Este descriptor utiliza la siguiente técnica:

Para determinar los ejes $\{u,v\}$ que representen un eje de referencia local se hace un histograma de las orientaciones del gradiente dentro de la región. El histograma contiene sólo 6 celdas. Para calcular el descriptor SURF, se consideran los gradientes dx paralelos al eje u , y los gradientes dy paralelos al eje v . Se usan 4×4 subregiones. Una vez obtenidos los gradientes, se genera un vector $4 \times 4 \times 4 = 64$ casillas considerando 4 direcciones principales para cada región. Para rellenar el vector se utiliza un algoritmo que tiene en cuenta la magnitud del gradiente y su ángulo de inclinación. La creación de los descriptores en esta técnica se basa en sumas bidimensionales de los vectores obtenidos de los puntos de interés, a diferencia de la técnica SIFT que realizaba sumas tridimensionales.

El sistema SURF es robusto a cambios de iluminación, puntos de vista y es más rápido de calcular que el SIFT original, pero además de no ser invariante a deformaciones rígidas o elásticas, tampoco lo es a rotación.

2.1.3.3 Proyección de imágenes

La elección de la proyección en la que se presentarán las imágenes es el siguiente paso a realizar en la generación de la panorámica. Consiste en proyectar todas las imágenes en una sola, de forma que se vea desde un punto de vista correcto. Existen distintos tipos de proyecciones los más comunes son los siguientes:

- **Plana:** En las cuales no hay corrección de perspectiva.
- **Cilíndrica:** Un panorama cilíndrico está destinado a ser visto como si se curvara por el interior de un cilindro.

- **Esféricas:** Las imágenes esféricas generalmente están representados de forma equirrectangular y representan 360 grados sobre el eje horizontal y 180 grados sobre la vertical.

2.1.3.4 Fusión de imágenes

El procedimiento de fusión de las imágenes es el proceso por el cual se combinan múltiples imágenes para producir una única imagen panorámica o una imagen de alta resolución evitando en la medida de lo posible que se distinga los puntos de unión.

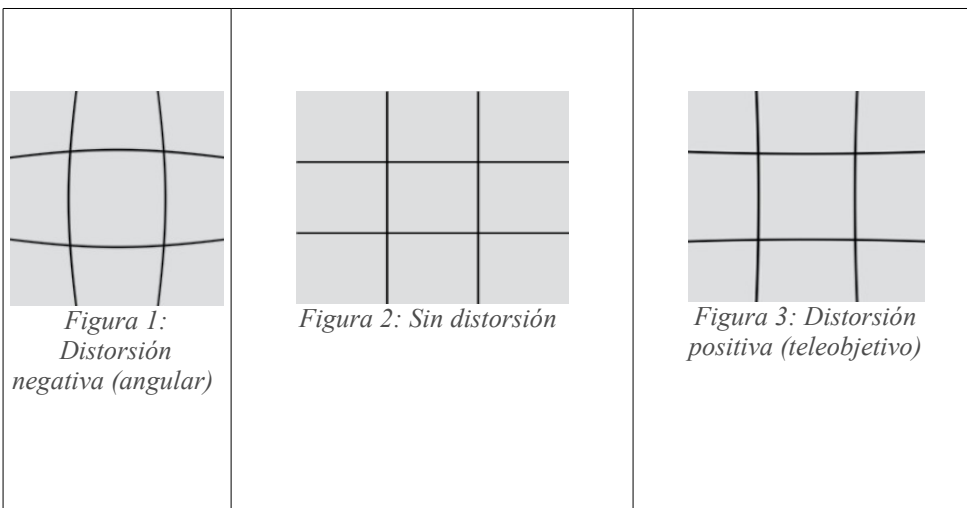
El proceso de fusión de imágenes está compuesto por un conjunto de fases claramente diferenciadas.

- Calibración de imágenes.
- Transformación de imágenes.
- Mezcla de imágenes.

2.1.3.4.1 Calibración de Imágenes

En la que se deberá realizar la corrección de **perspectiva**, corrección de barril, corrección de **viñeteado** (vignetting) ,corrección de la aberración **cromática**, etc..

- Cuando realizamos una fotografía, dependiendo de la lente, se nos produce un problema que es la deformación de la imagen. Es lo que se conoce como distorsión, esta puede ser positiva o negativa.



- La **corrección de perspectiva** permite solucionar problemas que aparecen en las fotografías realizadas en las cuales se pueden ver efectos como el que se ve en la siguiente imagen en la que los elementos de la parte superior tienden a desplazarse del centro natural apareciendo como inclinados cuando realmente deberían aparecer rectos.



Figura 4: Corrección de perspectiva vertical

- La **distorsión de barril** es un efecto producido por muchos objetivos que hace que los objetos queden redondeados, es especialmente visible en la línea del horizonte o los bordes de los edificios.



Figura 5: Efecto Barril

- El **vignetting** en fotografía se presenta en la aparición de sombreados con pérdida casi total de información en los bordes de las imágenes. El viñeteo consiste en el oscurecimiento y por consiguiente una mayor saturación de color de una fotografía en sus cuatro esquinas producida en su mayor parte por un **menor** aporte de luz en la zona más periférica del círculo de visión [VIGNETTING].
 - Este efecto se produce debido a varias causas y cuanto mayor es la distancia focal (teleobjetivos) menos se nota el efecto. Este problema también se nota mucho más en las cámaras digitales pues los CCD son menos tolerantes a estas pequeñas diferencias de intensidad de luz que las películas.



Figura 6: Efecto vignetting

- La aberración **cromática** (o perturbación del color) se debe principalmente a que la luz a su paso a través de una lente sufre una desviación. Esta desviación no es la misma para cada longitud de onda o color (El azul es el que tiene menor longitud de onda). Cuando no todos los colores convergen en un mismo punto, en este caso, en la película o sensor, aparece la aberración cromática o de color. El efecto en las fotos se manifiesta en ciertos contraluces en los que aparezcan halos de color azul o púrpura rodeando la figura.



Figura 7: Efecto Aberración cromática (Azul)

- Las consecuencias de las aberraciones cromáticas se minimizan trabajando a aberturas pequeñas, porque así todos los rayos luminosos se ven obligados a atravesar el objetivo por el centro, donde la curvatura y las diferencias de refracción son mínimas.

2.1.3.4.2 Transformación de imágenes

Previo a la construcción de la panorámica es preciso realizar una serie de modificaciones en las características de las imágenes para solucionar los problemas anteriormente comentados. Las herramientas analizadas disponen de mecanismos para rotar las imágenes, compensar la exposición, etc.

2.1.3.4.3 Mezcla de imágenes

En el proceso de mezcla de imágenes es necesario realizar un conjunto de adaptaciones para conseguir el resultado esperado, que son las siguientes:

2.1.3.4.3.1 Corrección del color

El objetivo de esta técnica es intentar hacer coincidir las áreas adyacentes procedentes de imágenes distintas en color, contraste y brillo. Existe software como *enblend* que permiten "ecualizar" el tono y brillo de las fotos evitando que se noten por donde se han pegado.

2.1.3.4.3.2 HDR

En algunos casos en el proceso de generación de la panorámica interesará generar un formato extendido HDR.

HDRI, o ‘**High Dinamic Range Imaging**’, no es más que un procedimiento para recoger en una sola imagen todo el rango de luminosidades posibles, y no sólo las que es capaz de detectar el ojo humano, sencillamente todas las variables existentes en un entorno, para poder manejar y representar un modelo fiel y realista de la propia realidad

Actualmente lo común es tratar con imágenes con una profundidad de color de 8 bits para cada píxel y canal (los canales rojo, verde y azul, que juntos recomponen la imagen a pleno color). Pero 8 bits no son suficientes para representar toda la gama de luces y colores que es capaz de detectar el ojo humano en pleno uso de su capacidad, pues éste es capaz de detectar negros mucho más negros que el color más oscuro representable en una pantalla así como luces muchísimo más intensas. 256 valores significan que el blanco más blanco, en ese caso, sería sólo 256 veces más luminoso que el negro más negro.

La forma factible de representar esto es aumentando la profundidad de bits por canal hasta los 32 bits, lo que otorga más de 4 mil millones de valores posibles para cada píxel. Esta ‘profundidad de color’ va a permitir finalmente, representar y guardar con absoluta fidelidad todos los matices de la imagen, desde la sombra más oscura hasta la luz más brillante con un detalle asombroso: más real que la propia realidad.

2.1.3.4.3.3 Compensación de movimiento

En determinados tipos de panorámicas es necesario realizar un filtro para compensar el movimiento de determinados tipos de objetos que se mueven, por ejemplo nubes. Para solucionar esto existen diferentes filtros que permiten eliminar el efecto “*ghost*” de la imagen resultado por ejemplo Celeste [CELESTE].

2.2 Herramientas mosaicos

En esta sección se describen las aplicaciones software que se han ido evaluando durante el desarrollo del trabajo centrándonos en las posibilidades que tienen para implementar las características anteriormente comentadas tanto a nivel de composición de panorámica como para solucionar problemas que puedan tener las imágenes.

Aunque se ha pretendido centrar el estudio de herramientas a aquellas **de código abierto** y disponibles para un entorno GNU/Linux es importante conocer las posibilidades que hay en el mercado. En las siguientes secciones se explican las características más importantes de las aplicaciones evaluadas.

2.2.1 Comerciales

2.2.1.1 AutoStitch

Características

- Muy sencillo. Realiza el panorama de forma automática.
- En caso de ser necesario, hay que rotar las imágenes antes de ensamblarlas.
- No permite modificar parámetros ni recortar la imagen resultante.
- Sólo proyección cilíndrica.

Disponibilidad

- Sólo para Windows
- Software propietario; gratuito pero no libre

Descarga.

- <http://people.cs.ubc.ca/~mbrown/autostitch/autostitch.html>

En la Figura 8 se puede ver una pantalla de ejemplo del programa.

2.2.1.2 Autopano Pro

Características

- Interfaz muy intuitivo y fácil de utilizar.
- Realiza búsquedas automáticas de panoramas entre un conjunto de fotos.
- Es muy preciso a la hora de ensamblar las imágenes, pero no acepta la incorporación de puntos de control de forma manual.
- Soporta distintas proyecciones.
- Permite corrección de color.
- Soporta formato HDR.
- Permite exportar en formato multicapa y TIFF.

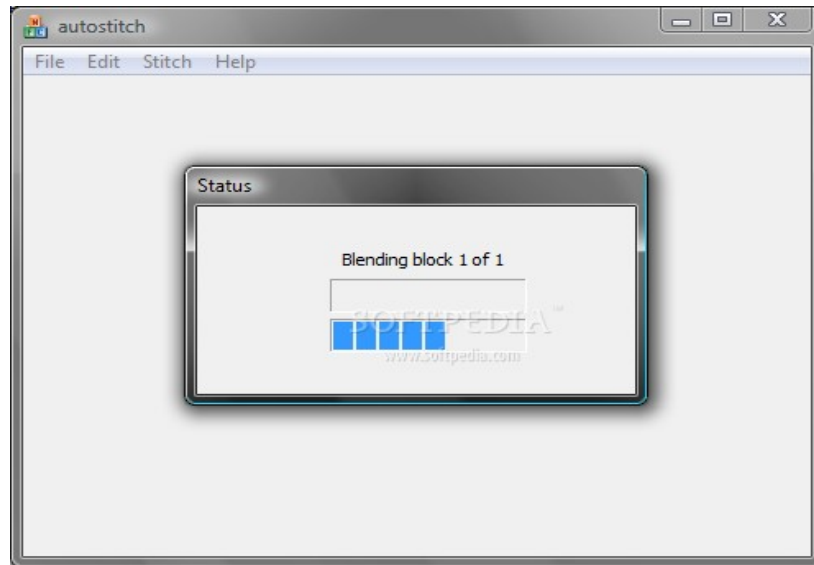


Figura 8: AutoStitch ScreenShot

Disponibilidad

- Disponible para Linux, Windows y Mac
- Software propietario con coste de licencia.

Descarga.

<http://www.autopano.net>

2.2.1.3 PTGui

Características

- Rapidez. Tarda muy poco tiempo en calcular los puntos de control y mostrar la previsualización.
- Precisión. Aceptando la incorporación de nuevos puntos de control de forma manual, al igual que Hugin (ver punto 2.2.2.1).
- Soporta imágenes tipo HDR
- Distintas proyecciones
- Permite exportar en formato multicapa y TIFF

Disponibilidad

- Disponible para Windows y Mac
- Software propietario con coste de licencia.

Descarga.

- <http://www.ptgui.com>

En la Figura 9 se puede ver una pantalla de ejemplo del programa.

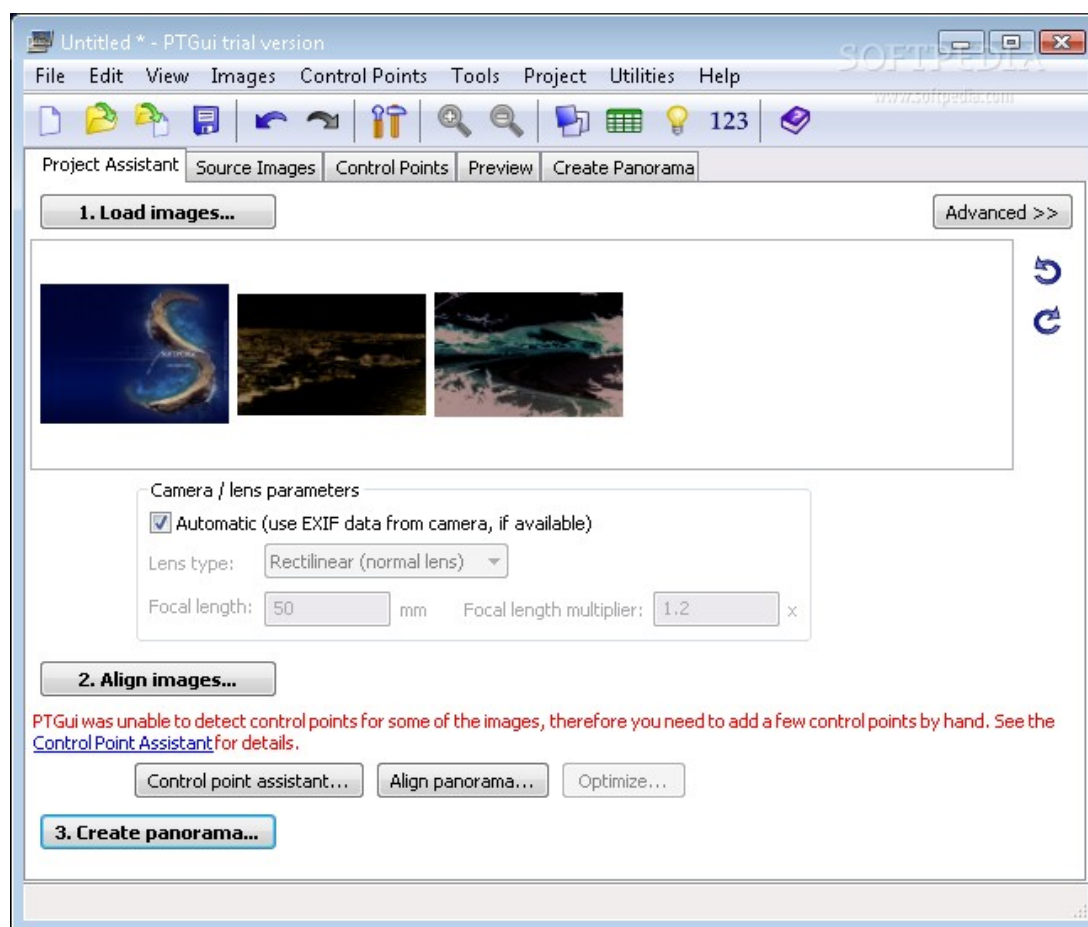


Figura 9: PTGui

2.2.1.4 PanoWeaver

Características

- Ajuste FOV, Pan, inclinación para cada escena
- Ensamble imágenes en formatos y formas con compatibilidad universal
- Elija para publicar su panorámica entre JAVA o QuickTime.
- Auto ensamblaje

Disponibilidad

- Disponible para Windows
- Software propietario con coste de licencia.

Descarga.

- <http://www.easypano.com>

En la Figura 11 se puede ver una pantalla de ejemplo del programa.

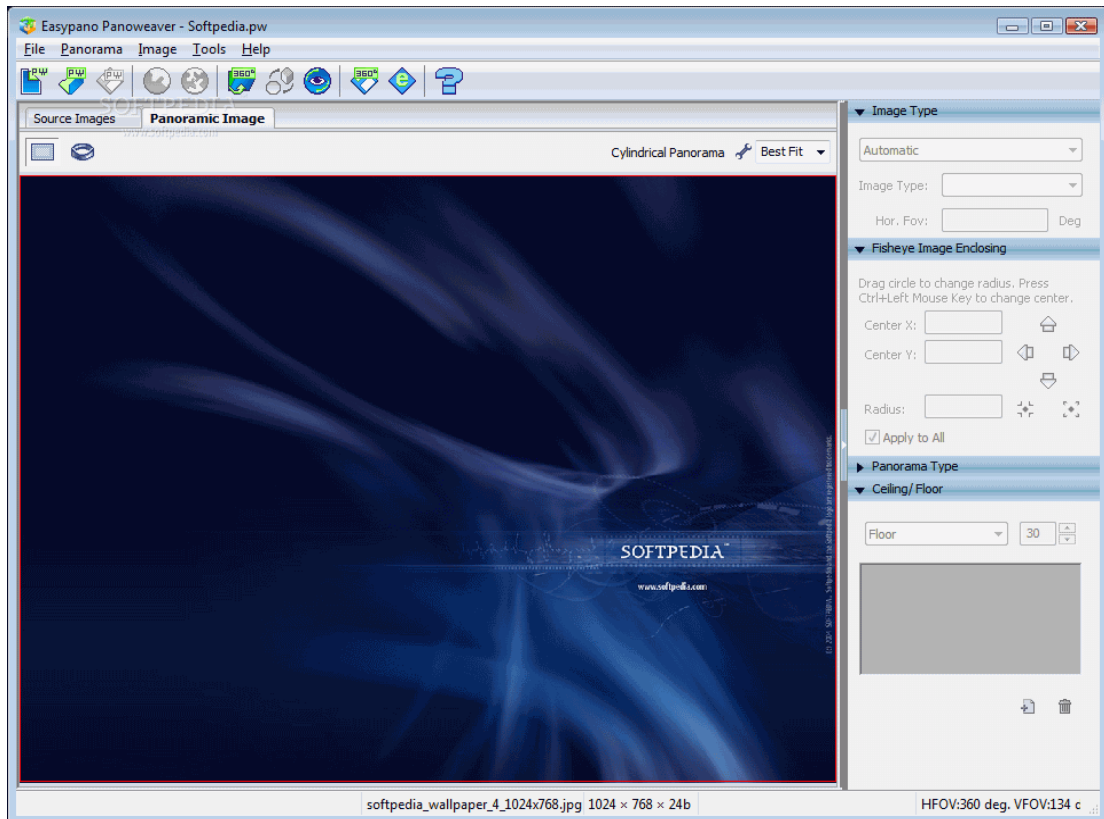


Figura 11: EasyPano ScreenShot

2.2.1.5 Otros

Existen un sinfín de herramientas preparadas para generar panorámicas pero no se ha realizado un estudio de las mismas. Entre estas herramientas están

- VRToolBox
- RealViz
- Panorama Tools
- PhotoShop CS3
- photosynth

2.2.2 Código abierto

La mayoría de los programas opensource analizados se basan en las panorama *tools* por lo que construyen su interfaz a partir de las funcionalidades que ofrecen estas librerías.

2.2.2.1 Hugin

Características

- Tarda algo de tiempo en calcular los puntos de control, ya que reajusta el tamaño de las imágenes.
- Precisión. Acepta la incorporación de nuevos puntos de control de forma manual.
- Soporta imágenes tipo HDR
- Maneja distintas proyecciones
- Permite exportar en formato TIFF

Disponibilidad

- Disponible para Linux, Windows y Mac
- Software libre

Descarga

- <http://hugin.sourceforge.net>
En la Figura 12 se puede ver una pantalla de ejemplo del programa.

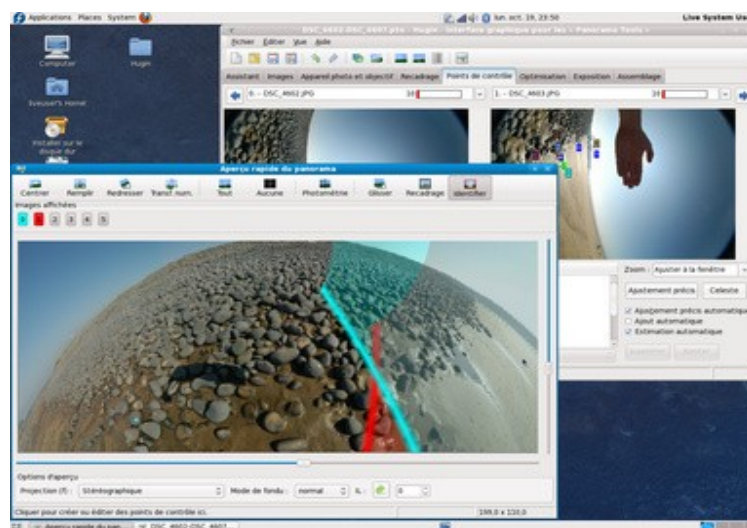


Figura 12: Hugin ScreenShot

2.2.2.2 PanoPoints

Características

- Soporta puntos horizontales y verticales.
- Permite mover los puntos de una forma sencilla
- No se mantiene ya que ha sido sustituido por hugin.

Disponibilidad

- Disponible para Linux, Windows y Mac (Necesita Perl y GTK)

- Software libre

Descarga

- <http://panopoints.sourceforge.net/>

En la Figura 13 se puede ver una pantalla de ejemplo del programa.

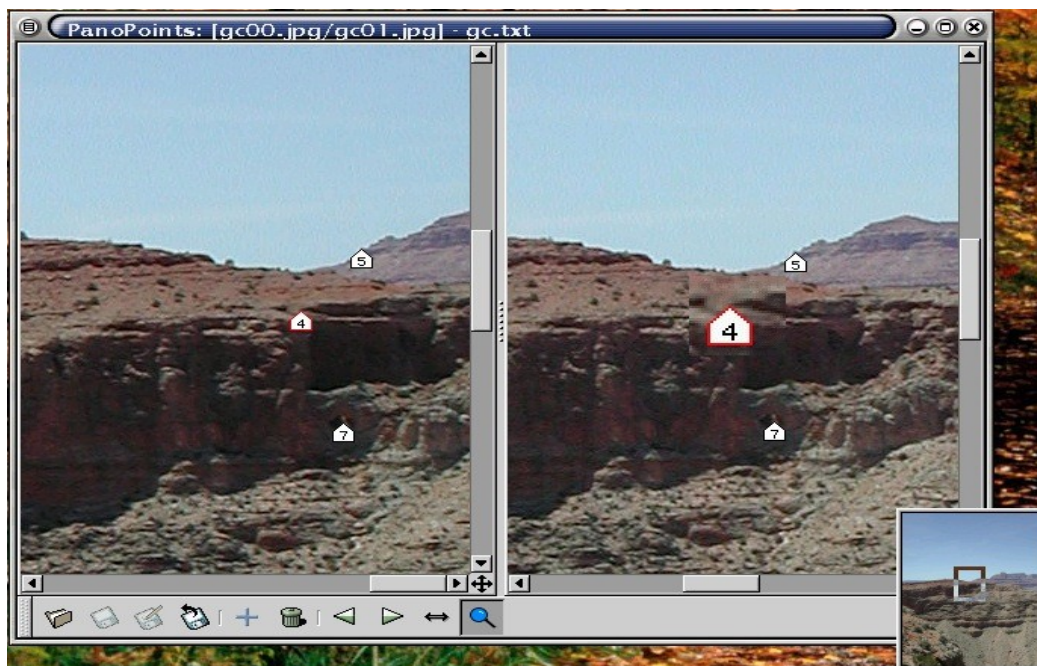


Figura 13: PanoPoints ScreenShot

2.2.2.3 Panorama Tools Open GUI

Características

- Soporte puntos horizontales y verticales.
- Permite usar programas externos de la *suite* imagemagick para convertir imágenes.
- No se mantiene ya que ha sido sustituido por hugin.

Disponibilidad

- Disponible para Linux, Windows y Mac (Necesita Perl y GTK)
- Software libre

Descarga

- <http://sourceforge.net/projects/ptbcbgui/>

En la Figura 14 se puede ver una pantalla de ejemplo del programa.

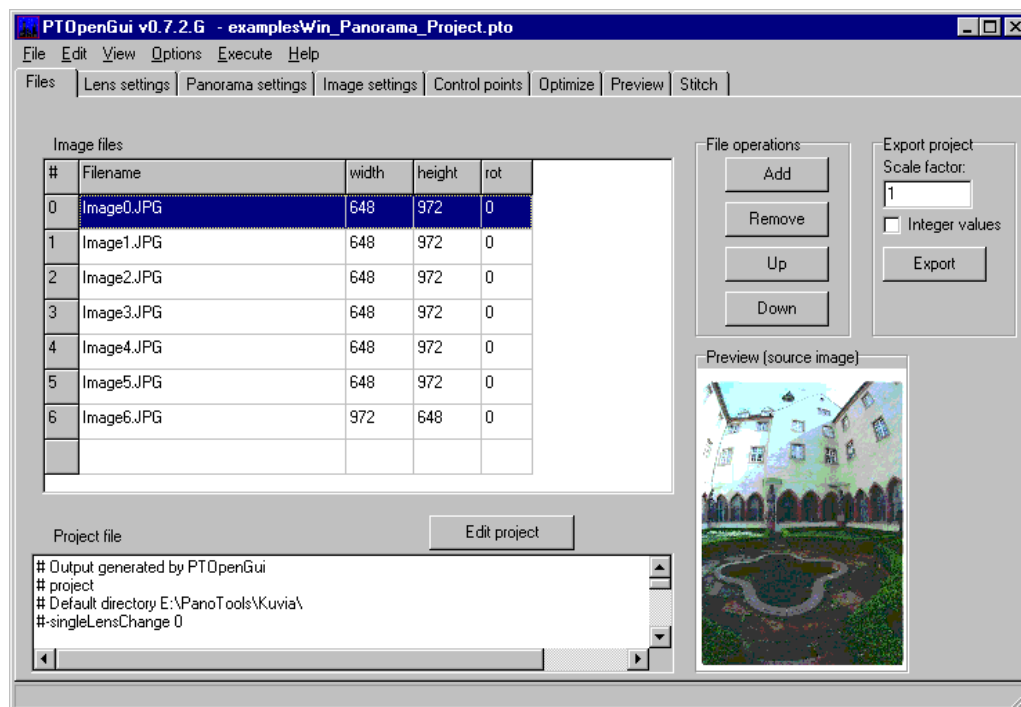


Figura 14: Panorama Tools Open Gui ScreenShot

2.2.2.4 iMerge

Características

- Permite alineamiento y control de brillo automático.
- Destinado para astromosaicos.
- Soporte RGB.
- Herramienta para ajuste de histogramas.

Disponibilidad

- Disponible para Windows
- Software para uso no comercial.

Descarga

- <http://jaggedplanet.com/iMerge.html>

2.2.2.5 AstroMosaic

Características

- Implementación GPL de iMerge
- Permite alineamiento y control de brillo automático.

- Destinado para astromosaicos.
- Soporte RGB.
- Herramienta para ajuste de histogramas.

Disponibilidad

- Disponible para Windows
- Licencia GPL

Descarga

- <http://christophe.jalady.free.fr/astromosaicgui/index.html>

2.2.2.6 Otros

Más que herramientas típicas de generación de mosaicos existen una serie de programas gratuitos que permiten realizar parte de las fases del proceso de alineación.

- PTPicker: Aplicación java para crear puntos de control que posteriormente pueden ser usados en otras aplicaciones.
- ControlPoints: Herramienta para ajustar los puntos de control permitiendo el moviéndolo
- PanoPoints: Similar a ControlPoints.
- PanoWizard: *Front End* de las Panorama Tools que funciona de una forma similar al hugin.

2.2.3 Conclusiones

El análisis y las pruebas realizadas con los distintos programas han mostrado resultados bastante dispares. De entre las herramientas analizadas independientemente de su disponibilidad y gratuidad dos de ellas han dado unos resultados bastante interesantes.

La primera **AutoPano Pro** una herramienta de pago pero con un interfaz muy sencillo e intuitivo. La mayoría sino la totalidad de las pruebas realizadas con la misma han dado unos resultados muy buenos. De hecho los valores obtenidos en el proceso de generación de la panorámica (distancia focal, factor de recorte, HFOV) han sido utilizados para otros programas como base para la generación de las panorámicas con resultados también interesantes.

La otra herramienta es **hugin** con una interfaz muy similar al AutoPano Pro pero open-source y que consigue también unos resultados muy interesantes. Aunque dispone de una interfaz también bastante intuitiva con un asistente incorporado dispone de muchas más opciones a la hora de realizar la panorámica por lo que en algunos casos no está tan sencillo como el AutoPano.

La ventaja principal de hugin además de que es un producto respaldado por la comunidad open-source y con una evolución continua durante los últimos años, es el acceso al código fuente lo que permite estudiar las funcionalidades y el mecanismo de integración las librerías

que utiliza con el fin de poder integrarlo en la aplicación desarrollada.

2.3 Proyecto Ciclope

En la siguiente sección se enmarca el software de mosaicos dentro del proyecto Ciclope con el objetivo de conocer cual sería su utilidad dentro de la plataforma.

Ciclope es un sistema que pretende resolver el problema de la falta de clases prácticas en carreras y escuelas técnicas. Esto es debido a varias razones como la falta de recursos, el acceso a los laboratorios está restringido, bien por que no se dispone de personal suficiente, etc.

La forma de resolver estos problemas es el empleo de laboratorios de acceso remoto vía Internet, concretamente a través de Web. De esta forma se puede ofrecer un recurso durante 24 horas al día, todos los días del año, y con una política de acceso definida por administradores. Con este enfoque, un usuario sólo necesita un ordenador con conexión a Internet y un navegador. De esta forma se pueden reducir las necesidades de los potenciales usuarios, reducir costes en los laboratorios y aumentar el rendimiento de las instalaciones. Para permitir el uso remoto también se desarrollan herramientas que permiten la gestión de recursos y colaboración dentro de los laboratorios.

2.3.1 Módulos

El sistema Ciclope esta compuesto por módulos, denominando así a cada uno de los componentes que conforman el sistema. El núcleo central del sistema se denomina Ciclope Core y es el único módulo estrictamente necesario para el uso del sistema, gestiona usuarios, bases de datos y partes comunes evitando problemas de redundancia de datos.

Cada módulo es independiente de los demás módulos lo que implica que el administrador pueda elegir que módulos pueden ser instalados y cuáles no. La instalación de un nuevo módulo produce una modificación en la configuración del módulo Core relacionada con las bases de datos y páginas de acceso.

En la Figura 15 se puede ver la arquitectura del proyecto.

Los módulos principales de este sistema son los siguientes:

- **Core**: Es el módulo central, al cual se conectan e integran el resto de módulos desarrollados. Dentro de éste se hace la siguiente división interna:
- **Laboratorios**: Son escenarios remotos que permiten teleoperar con un laboratorio físico.
- **Herramientas**: Son utilidades varias, algunas de las cuales son comunes a diversos laboratorios y otras se pueden utilizar de forma independiente al sistema Ciclope.
- **Documentación**: Trabajos o estudios de investigación relacionados de algún modo con Ciclope.

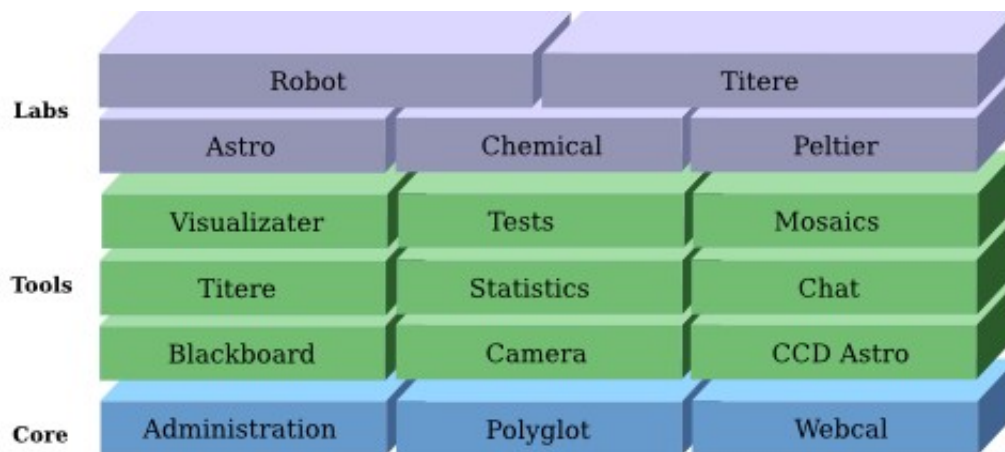


Figura 15: Estructura proyecto Ciclope

Para más información sobre este módulo ver [CICLOPE09] y [CICLOPE07]

El trabajo realizado en este proyecto se ubica dentro del módulo de Herramientas y permitirá explotar la información obtenida a través del Módulo Ciclope Astro el cual proporciona una serie de herramientas para experimentos astronómicos, creación de escenarios de este tipo y control de herramientas tales como telescopios y cámaras.

2.4 Aportaciones del TFC

Con este proyecto se pretende acercar a los usuarios la tecnología de generación de mosaicos a través de una aplicación Web accesible desde cualquier navegador y desde cualquier sistema operativo.

Una de las aportaciones de este TFC se basa precisamente en el uso de la tecnología Web para permitir que los usuarios puedan ver las posibilidades que tienen los productos de generación de mosaicos. El estar accesible a través de un navegador hace mucho más sencillo poder ver los resultados de una forma inmediata y permite que las imágenes generadas puedan en algún momento ser compartidas para que otros usuarios puedan disfrutar de las composiciones generadas, ya que el concepto de colaboración entre usuarios es una de las bases de proyecto.

El proyecto sigue la filosofía del código libre y cuanto sea desarrollado se hará público bajo la licencia GNU GPL, garantizando la posibilidad de crecer con nuevas aportaciones y mejoras. El código está disponible en Sourceforge en la siguiente URL [c](#)

De la misma forma se aporta un API de programación que puede ser utilizado desde otros tipos de aplicaciones y permite que se puedan extender las funcionalidades de las mismas con nuevas aportaciones. Este API permitiría por ejemplo que los usuarios pudieran integrar el proceso de composición de mosaicos en sus aplicaciones sin hacer uso de la aplicación Web proporcionada como producto del proyecto desarrollado.

3 Descripción del trabajo

En el presente capítulo se va a realizar una descripción del trabajo previo al desarrollo de este proyecto fin de carrera.

El trabajo realizado se ha llevado a cabo en un conjunto de fases claramente diferenciadas con el objetivo de disponer como hito final de un producto software completo que permitiera a un usuario llevar a cabo el proceso de generación de imágenes panorámicas a través de una aplicación web.

En los siguientes apartados se realizará un análisis de la integración realizada comenzando con la planificación detallada del proyecto, los pasos iniciales realizados y un análisis de la integración que se ha llevado a cabo.

3.1 Planificación

La planificación global del proyecto se puede ver en la siguiente ilustración. Esta planificación se ha modificado durante el avance del proyecto para reflejar los problemas encontrados los cambios detectados no contemplados al principio del proyecto, etc.

Como se puede ver en la Figura 16 los dos primeros meses se han dedicado al estudio de aplicaciones que permitieran ser usadas para integrarlas dentro de un entorno web.

Las pruebas realizadas consistieron en testear diferentes aplicaciones usando diferentes tipos de fotografías para ver los resultados producidos.

3.2 Primeros pasos

3.2.1 Premisas de partida

Como premisa de partida, como se puede ver más adelante en el apartado 4.1 de requisitos de usuario, el software se debería ejecutar en un entorno específico y disponer de al menos las siguientes características.

- Entorno Unix, Linux (Debido a que el sistema Ciclope se enmarca dentro de este entorno).
- Carente de licencias comerciales, por tanto un producto opensource.
- Permitirá que un usuario pueda hacer uso del mismo desde un navegador evitando tener que descargarse ningún tipo de software en su PC.
- Facilidad de uso.

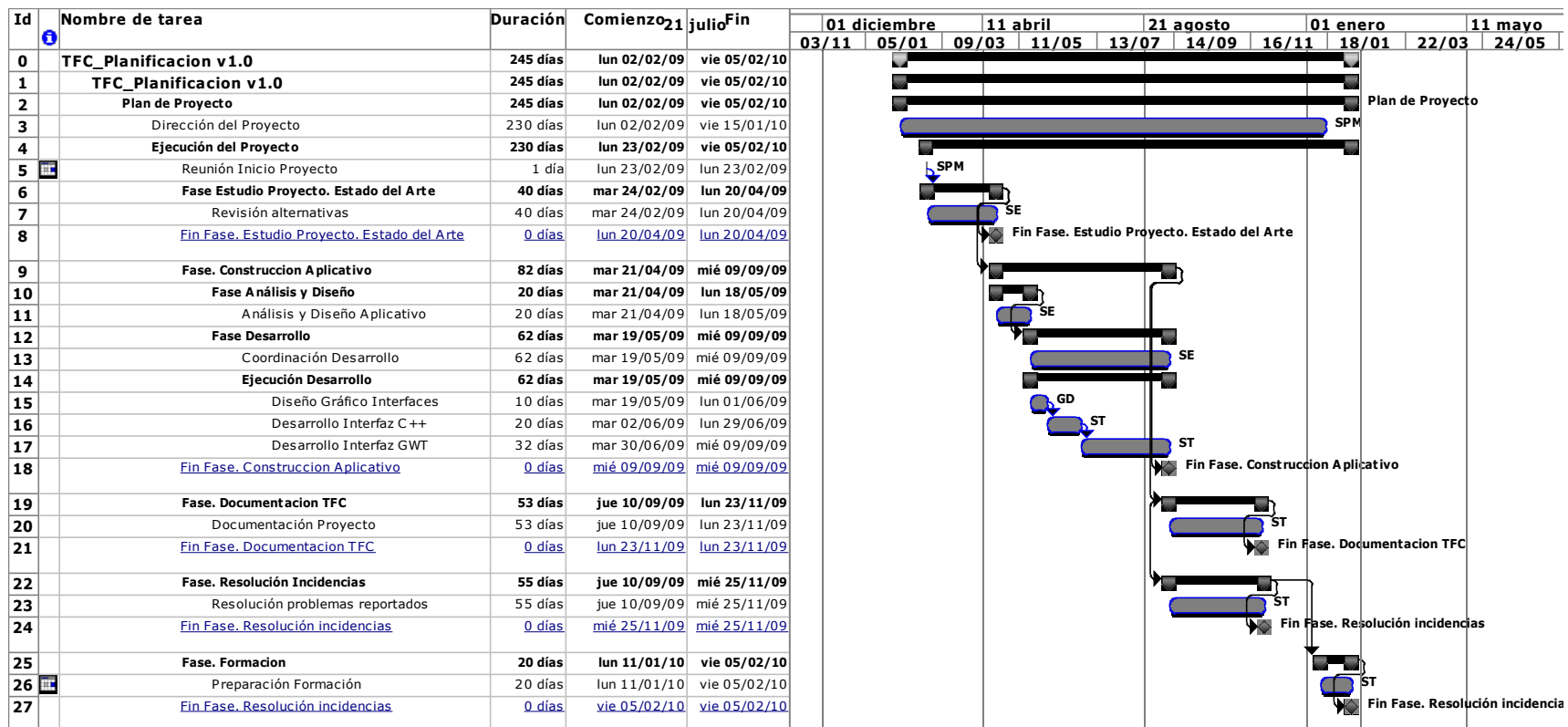


Figura 17: Planificación

Estas premisas se han tomado como base y el esfuerzo de esta primera fase se ha centrado en la búsqueda de productos con estas características. No obstante en el mercado como se ha podido ver en el apartado de Estado del Arte existen muchas herramientas que no disponen de estas características pero que consiguen unos resultados muy prometedores.

La mayoría de las herramientas analizadas disponen de asistentes que permiten que el proceso de generación sea muy sencillo e intuitivo para el usuario. Estos asistentes aíslan en cierta forma al usuario de tener conocimiento sobre términos y tecnología fotográfica.

Este aislamiento en los asistentes para usuarios no avezados se complementa con opciones avanzadas en la inmensa mayoría de los programas, las cuales permiten extraer toda la potencia de los mismos, permitiendo la generación de mosaicos muy complicados y espectaculares.

Para estos usuarios avezados el conocimiento de la tecnología de la fotografía digital, sus términos y sus técnicas permite un mejor conocimiento del proceso de generación.

3.2.2 Fotografía digital

Llegado el momento en este punto del trabajo y analizando la cantidad de términos fotográficos existentes en los diversos programas analizados se estimó que para comprender mejor el proceso era necesario tener un conocimiento básico en esta tecnología. Existe en Internet una gran cantidad de información sobre tecnología fotográfica así como numerosos foros donde se pueden compartir experiencias con otros usuarios en esta materia.

En el apéndice Apéndice 2. Fotografía digital se da a conocer al usuario una base fotográfica básica, con los términos que posteriormente se han tomado como base para el desarrollo del aplicativo. No obstante este trabajo no es específico sobre fotografía digital por lo que se ha intentado no realizar una descripción muy exhaustiva, objetivo quizá de otro trabajo más especializado sobre este tema.

Una vez conocido más en detalle que características dispone una fotografía y que factores afectan en el momento de la toma, es más sencillo o por lo menos mucho más evidente conocer los parámetros que afectan a las composiciones panorámicas. No obstante sí es preciso comentar que los programas analizados disponen de opciones tan avanzadas que sería preciso un capítulo completo para su análisis.

De entre todos los programas analizados el que disponía de características más idóneas para su uso dentro de la aplicación y el que permite que el usuario pueda personalizar la composición hasta límites insospechados es el que se ha comentado en el apartado de Estado del Arte, el programa Hugin [HUGIN].

3.2.3 Elección del producto

Hugin, como se ha comentado anteriormente, es un front-end de las Panorama Tools [PANOTOOLS] una serie de librerías originariamente desarrolladas por el profesor Helmut Dersch y que fueron mantenidas en su sitio Web hasta el año 2002 fecha en la cual dejaron de distribuirse (parece ser que por un problema de patentes aunque no está claro del todo). En el año 2004 se distribuyó parte

del código fuente y la comunidad empezó a mejorar las librerías.

Concretamente se distribuyó el código fuente de la librería dinámica (Pano12) pero no de ninguno de los programas accesorios (PTStitcher, PTOptimizer,etc.) que tuvieron que ser rehechos desde cero (El sustituto del PTStitcher actualmente es el PTMender).

Estas librerías han evolucionado por tanto durante los últimos años y pretenden convertirse en un estándar en el proceso de generación de mosaicos.

Hugin es un programa que como el resto de programas analizados dispone de un asistente que permite un proceso inicial de aprendizaje en la herramienta en el que no es necesario conocer en profundidad las técnicas de composición.

El usuario selecciona una foto y el sistema analizar en busca de puntos comunes utilizando algoritmos matemáticos. Seguidamente si todo es correcto muestra al usuario una previsualización. Si la previsualización es del agrado del usuario éste en una última fase puede generar la panorámica final.

El usuario tiene la posibilidad en cada fase de personalizar la composición para lo cual tiene distintos paneles donde puede modificar los parámetros que afectan al proceso de composición.

Dispone de la posibilidad desde incluir nuevos puntos para mejorar la calidad de la panorámica. En las Figura 18.y Figura 19 se puede ver como se pueden llegar a modificar los valores actores de exposición, corrección de brillo, etc.

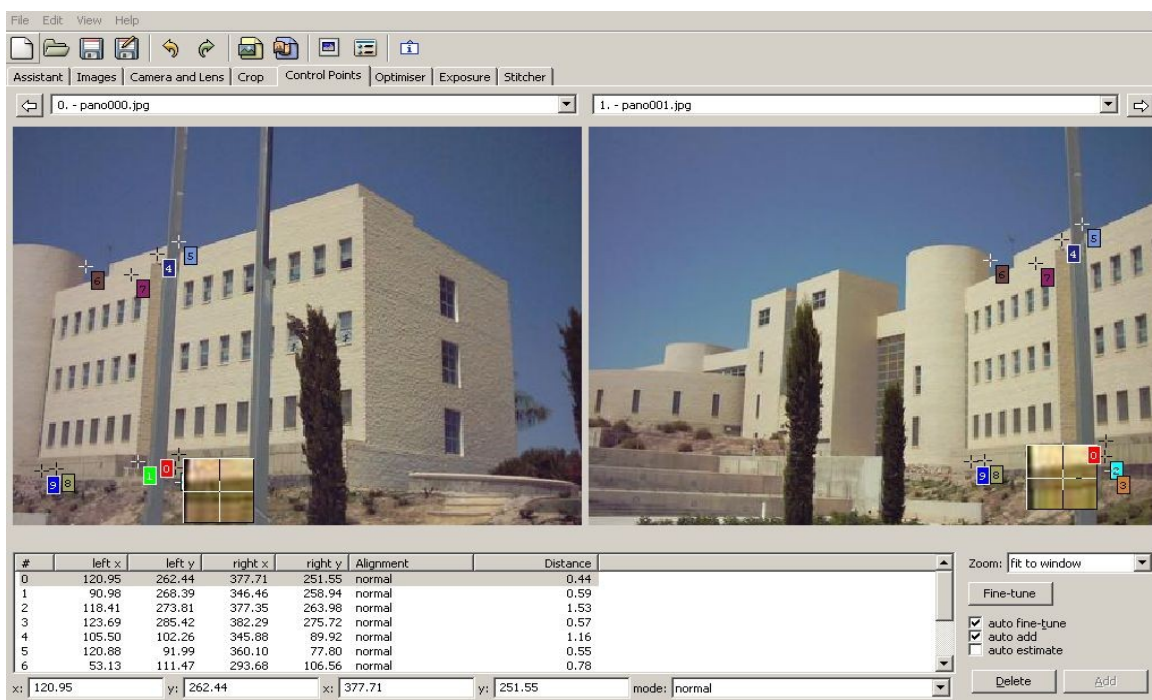


Figura 18: Hugin1. Inclusión de puntos

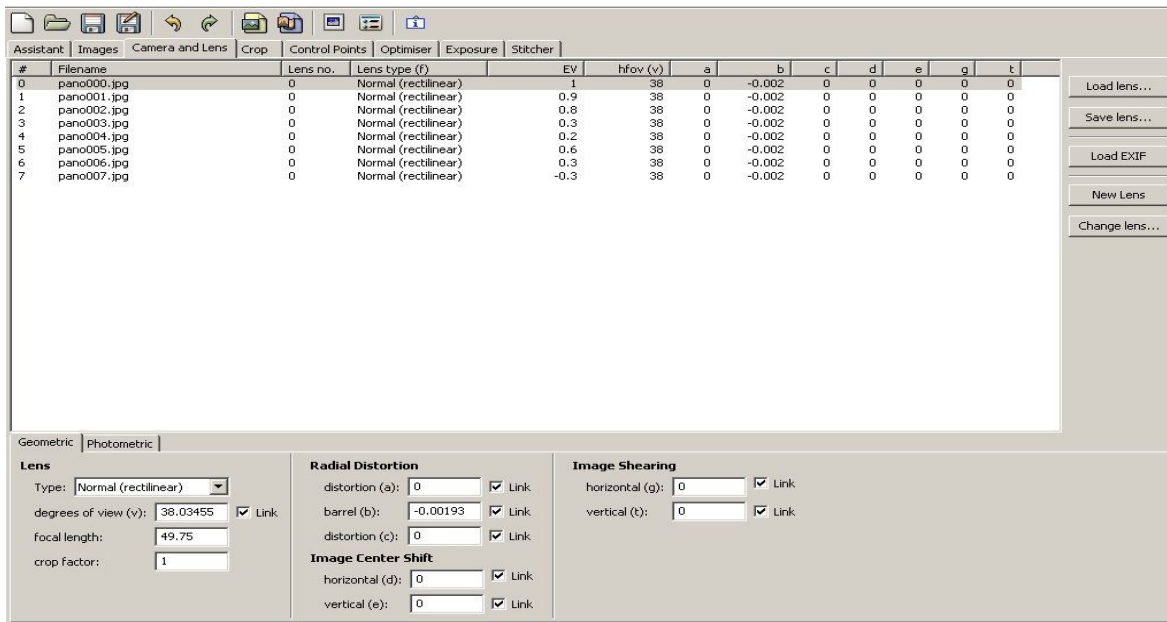


Figura 19: Hugin2

Tomando como base las características del software, su estructura y su algorítmica así como las premisas del tipo de software a realizar se comenzó a estudiar más en detalle este software.

3.2.3.1 Características

Como se ha comentado anteriormente hugin es un frontend para un conjunto de librerías y programas denominado panorama tools. Este software está desarrollado en el lenguaje de programación C++ utilizando librerías de interfaz gráfica, concretamente wxWidgets [WXWIDGETS] que permiten ayudar al programador en la inclusión de botones paneles y en el manejo de eventos.

El uso de este lenguaje de programación en la herramienta provocaba en primer lugar un aumento en la complejidad del software a desarrollar, ya que el lenguaje de programación elegido para llevar a cabo los desarrollos era Java y la integración con C++ o C aunque posible presenta inconvenientes y como norma suele ser más compleja.

Una vez descargado el software y su código fuente [HUGINSRC] se empezó a analizar la estructura de paquetes y se pudo constatar que su estructura estaba muy bien conseguida y era muy clara permitiendo una lectura muy fácil del software.

Por las dependencias necesarias para compilar también se pudo observar que su funcionamiento estaba muy ligado a la utilización de al menos los siguientes componentes.

- libpano13
- PTblender

-
- PTmasker
 - PTmender
 - PTroller
 - libceleste
 - autopano-sift-c
 - enblend
 - enfuse
 - exiftool
 - nona
 -

Con respecto a las librerías dinámicas correspondían a funciones exportadas de la librería panotools mientras que los ejecutables necesario corresponden a programas complementarios a este software y que permiten ajustar el proceso de generación/optimización de mosaicos.

Por tanto hugin básicamente es un programa que explota las funcionalidades ofrecidas por estas librerías y ejecutables provocando un encadenamiento de funcionalidades con un resultado final que es el mosaico deseado. En la Figura 20 se puede ver la arquitectura del producto.

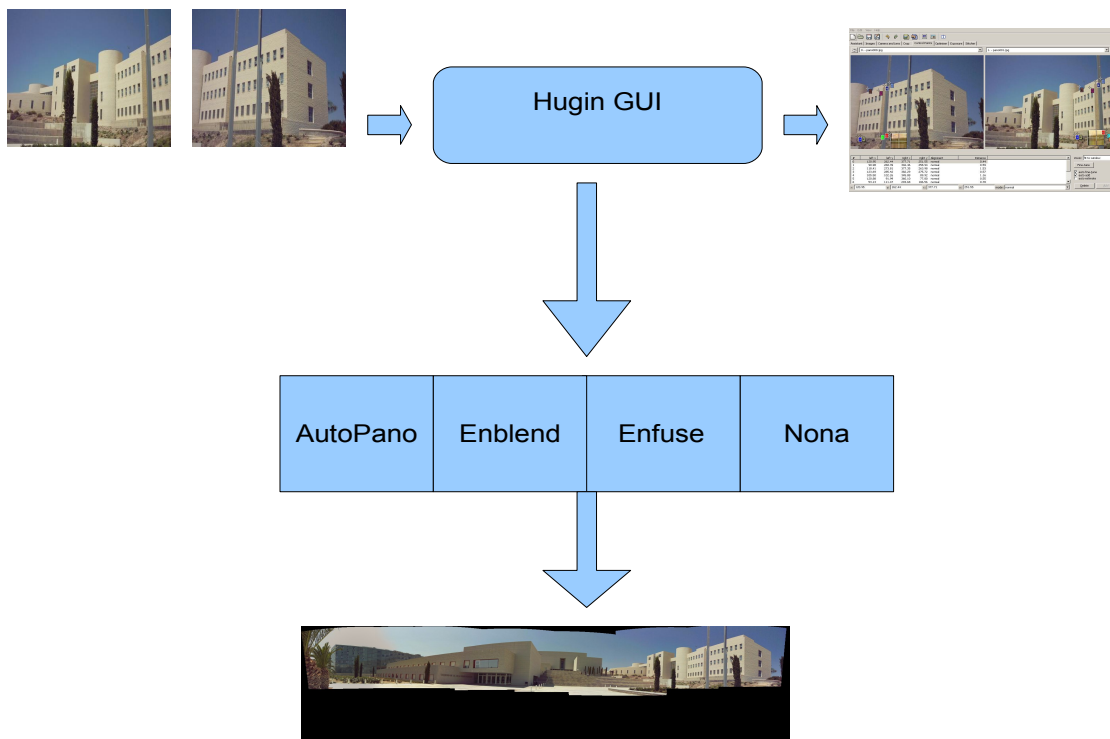


Figura 20: Arquitectura Hugin

3.2.3.2 Ventajas

A continuación se explican las ventajas que ofrece este software las cuales han sido claves en la elección del mismo.

- **Código libre:** El acceso al código y el estudio del funcionamiento han sido claves para comprender mejor el encadenamiento de servicios y su análisis para incorporarlo a la lógica de funcionamiento de la aplicación.
- **Basado en programas independientes:** Hugin no ofrece ni algoritmos, ni herramientas para generar imágenes ni nada relacionado con complicadas operaciones matemáticas las cuales son ofrecidas por programas que tienen una evolución en el tiempo y que en principio pueden mejorar sus funcionalidades ofreciendo mejores resultados al usuario
- **Desacoplamiento:** Hugin podría cambiar determinadas programas por otros que ofrezcan mejor rendimiento o más características simplemente manteniendo los interfaces.

3.2.3.3 Funcionamiento

El siguiente paso que se llevó a cabo fue extraer los interfaces de unión entre el software hugin y las librerías y programas para verificar cuanto de fuerte era ese acoplamiento. Esta fue quizá la tarea más complicada de todas porque era necesario realizar un análisis detallado del código fuente.

El objetivo era encontrar aquellos puntos que posteriormente pudieran ser utilizados o adaptados en un programa externo.

Lo que se buscaba también en estos interfaces era que no se dependiera de ninguna forma de librerías gráficas ya que esta interfaz iba a ser desarrollada utilizando otras tecnologías (**GWT**) que serán explicadas en el capítulo 5.

El procedimiento asíncrono también era un punto muy importante a tratar ya que las operaciones sobre todo la de análisis de puntos SIFT consumen muchos recursos y sobre todo tardan mucho tiempo en completarse por lo que es necesario que el usuario sea consciente en todo momento de que el proceso de generación está operativo y funcionando mediante indicaciones como barras de progreso o similares.

El funcionamiento del software hugin en su operativa más básica es el siguiente y se basa en el uso del asistente de generación el cual se puede completar con opciones más complejas de personalización.

- El usuario selecciona a través del asistente la lista de imágenes individuales que quiere procesar introduciendo si no está accesible en los metadatos de la imagen la información sobre el tipo de lente con la que el usuario tomó la imagen
- Seguidamente una vez que el sistema ha comprobado la idoneidad de las imágenes el usuario comienza el proceso de alineación de las imágenes buscando puntos comunes en las imágenes.
- Una vez que el sistema ha terminado buscando los puntos comunes se presenta al usuario dichos

puntos para que este pueda ver la idoneidad de los mismos. El usuario tiene a su disposición una vista preliminar en la que puede ver una previsualización de la composición.

- Una vez que el usuario ha ajustado todos los parámetros necesarios para optimizar la composición el siguiente paso es crear la panorámica final en la cual se realizan el ensamblaje de las distintas imágenes eliminados los puntos de cosido de las mismas y aplicando los filtros necesarios para solucionar problemas en la imagen final.

3.2.3.4 Suite Hugin

Hugin se integra con varios programas para llevar a cabo el proceso de generación de panorámicas. En algunos casos la integración es directa y se puede realizar directamente desde la interfaz de usuario mientras que otros programas disponen de una interfaz batch que permite corregir determinados parámetros de la imagen.

3.2.3.4.1 nona

- Este programa reemplaza al original PTSticher (originaria de la suite panotools) pero cuyo código fuente no se liberó. Actualmente ha aparecido otro aplicativo denominado PTMender que se distribuye en la Suite Panorama Tools y del que sí se distribuye el código fuente. Por lo tanto existen dos alternativas gratuitas (nona y PTMender).
- Básicamente su utilidad es la de realizar distorsiones geométricas y fotométricas de las imágenes, entre estas funcionalidades están el control de vignetting, balanceo de blancos, corrección de brillo, etc.

3.2.3.4.2 enblend

Enblend es una herramienta que se utiliza en el proceso final de generación de panorámica con el objetivo de eliminar o al menos minimizar los puntos donde se realiza la unión (costuras). Enblend se utiliza a modo de plugin desde el interfaz de usuario aunque también se puede utilizar desde línea de comandos.

3.2.3.4.3 Enfuse

Enfuse es una herramienta que se utiliza en el proceso final de generación de panorámicas con el objetivo de mejorar la exposición de la imagen.

Enfuse combina imágenes superpuestas utilizando el algoritmo Mertens-Kautz-Van Reeth [ENFUSE]. Este algoritmo permite de una manera rápida mezclar imágenes con diferentes niveles de exposición sin tener que recurrir a generar imágenes en formato HDR las cuales necesitan una mayor

capacidad de procesamiento y mayores capacidades de almacenamiento..

3.2.3.4.4 Fulla

Fulla es un programa con una interfaz batch a través de línea de comando con las capacidades de poder arreglar los siguientes problemas que se pueden producir en la toma de imágenes (distorsión de barril, aberración cromática y vignetting).

3.3 Análisis de la integración

Una vez seleccionado el producto y más concretamente las librerías con las cuales se iba a empezar a trabajar el siguiente paso que se llevó a cabo en el proyecto fue el de realizar el análisis y diseño de las características que tendría que tener el producto para que pudiera completarse correctamente el proceso de generación de mosaicos.

La integración de todos los componentes a desarrollar se realizó en varias fases cada una de las cuales dedicada a analizar, diseñar e implementar los distintos módulos de los que iba a estar compuesta la aplicación.

En la siguiente ilustración se puede ver el esquema modular del aplicativo y sobre el que se ha realizado todo el trabajo.

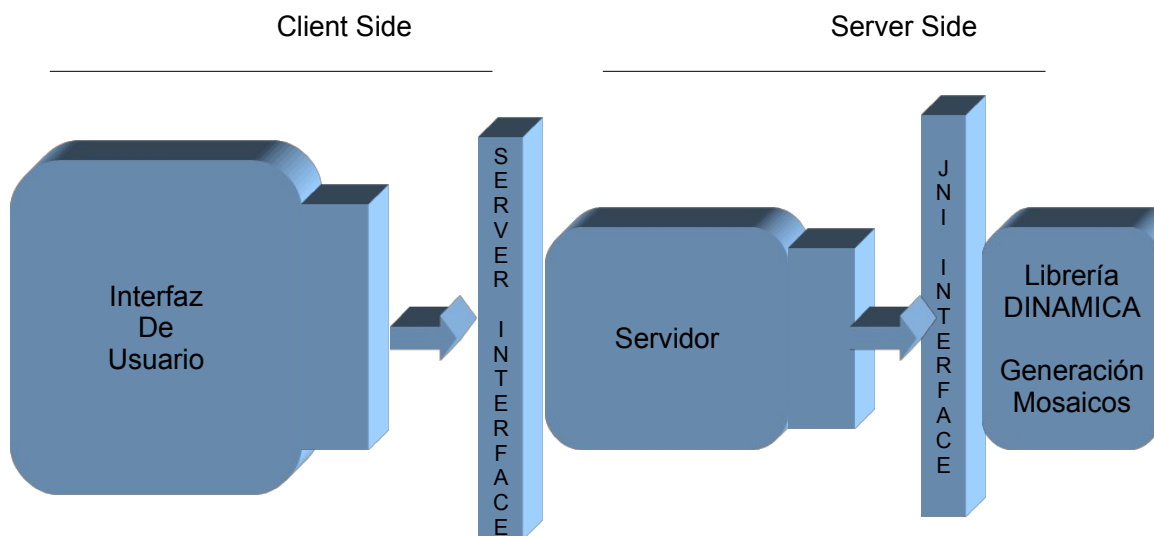


Figura 21: Arquitectura del aplicativo

3.3.1 Análisis de las librerías de composición

Tras el análisis de toda la información se consiguieron establecer una serie de interfaces con los que trabajar en el desarrollo de la aplicación, interfaces que posteriormente tendrían que ser desarrollados para poder ser invocados desde la aplicación

Esta definición de interfaces permite por ejemplo que se pueda usar esta librería dinámica generada desde programas externos al desarrollado en este trabajo.

Los interfaces que se extrajeron en este estudio fueron los siguientes:

- jniInit

-
- A través de esta interfaz se debe iniciar la librería dinámica para comenzar el proceso de composición.
 - JniLoad
 - A través de esta interfaz el usuario puede subir un conjunto de imágenes al servidor para su posterior proceso.
 - JniAlign
 - A través de esta interfaz el usuario comienza el proceso de alineación. Este método debe devolver inmediatamente el control a la aplicación que lo invoca ya que el proceso puede ser lento dependiendo del número de imágenes que se hayan subido al servidor.
 - JniStatusAlign
 - A través de esta interfaz el usuario puede verificar el estado del proceso de alineación.
 - jniCreatePanorama
 - A través de esta interfaz el usuario comienza el proceso de generación de la panorámica final. Al igual que el método que alinea las imágenes este método debe devolver inmediatamente el control al usuario.
 - JniStatusCreatePanorama
 - A través de esta interfaz el usuario puede verificar el estado del proceso de generación de la panorámica.

3.3.2 Análisis de la interfaz de usuario

El interfaz de usuario fue otro de los puntos clave en el desarrollo de la aplicación. El requisito fundamental de esta interfaz fue que el acceso a la aplicación se realizara a través de un navegador Web para que fuera uno de los módulos del proyecto Ciclope y con el objetivo de facilitar el acceso a los usuarios a la herramienta.

Los interfaces de usuario para los aplicativos Web accesibles desde un navegador han ido evolucionando durante los últimos años pasándose de aplicaciones muy sencillas y con muy poca interacción con el usuario hasta los últimos aplicativos muy robustos, más ligeros y con muchas más funcionalidades y capacidades de interacción con el usuario.

La Web 2.0 representa la evolución de las aplicaciones tradicionales hacia aplicaciones web enfocadas al usuario final. La evolución a esta tecnología se puede ver en los desarrollos que se están realizando para:

- Transformar software de escritorio hasta ahora solo disponible en este entorno hacia la plataforma del web.
- Separar el contenido que se presenta en las páginas Web del diseño de los mismos mediante el uso de hoja de estilos.
- Utilización de AJAX para manejar peticiones asíncronas.

•

Uno de los problemas principales que se encuentran los desarrolladores cuando trabajan con aplicaciones Web es la existencia en el mercado de varios navegadores Web que aunque intentan seguir las normas impuestas por el W3C [W3C] cada equipo de desarrolladores hace una interpretación de las mismas obteniendo como resultado una implementación diferente de la misma norma. De la misma forma otros navegadores intentan cubrir necesidades de usuarios no consensuadas en el W3C lo que lleva a implementaciones propietarias no compatibles con el resto de navegadores.

En la siguiente ilustración Figura 22 se puede observar el porcentaje de mercado a Septiembre de 2009 de los principales navegadores, estadísticas extraídas de [AULAMBRA]

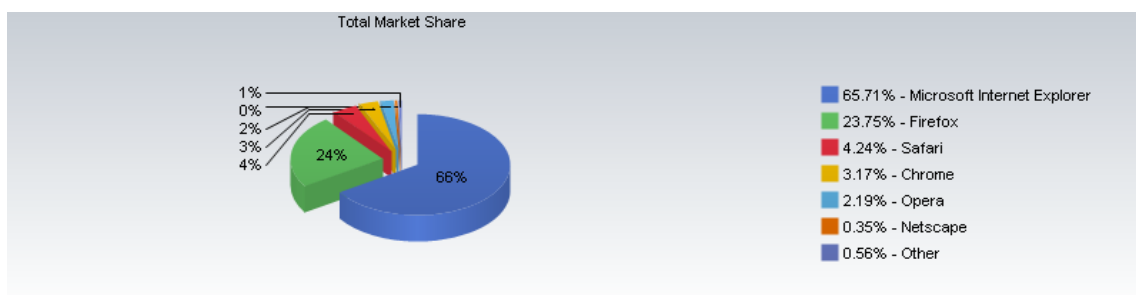


Figura 22: Porcentaje Uso Navegadores. Septiembre 2009

Actualmente debido a este y a otros factores, la creación de aplicaciones web resulta un proceso pesado y propenso a errores.

Los desarrolladores pueden pasar el 90% de su tiempo estudiando las peculiaridades de los navegadores. Por otra parte, la creación, la reutilización y el mantenimiento de una gran cantidad de componentes AJAX y bases de código Javascript pueden ser tareas complejas y bastante delicadas y sobre todo propensas a errores.

Es aquí donde surge un framework que pretende eliminar o al menos mitigar estos problemas. La tecnología se conoce como Google Web Toolkit (**GWT**) y es la que se ha elegido como base para el desarrollo de la aplicación.

Es un framework que se basa en la utilización del lenguaje **Java** para desarrollar las interfaces de usuario mediante la utilización de una serie de componentes disponibles a través de un API bastante completo. La utilización del lenguaje Java hace más accesible y sencillo la definición de los interfaces ya que es un lenguaje mucho más completo y sobre todo fuertemente tipado, esto quiere decir que al declarar una variable indicamos su tipo y no podrá cambiar de un tipo a otro automáticamente característica que no tiene Javascript en la que podemos incluir en una variable la información que deseamos, independientemente del tipo de ésta. Esta característica, aunque interesante para determinadas operaciones, hace más difícil leer el código y realizar un seguimiento del mismo produciendo bastantes problemas en el desarrollo.

Una vez desarrollado el interfaz Java GWT dispone de un compilador que genera el código resultado (HTML+Javascript) que está preparado para que pueda ser ejecutado en cualquier navegador. Este

compilador genera las adaptaciones necesarias para que las posibles diferencias entre navegadores estén enmascaradas dentro del código generado.

3.3.2.1 Primeros pasos

Para trabajar con GWT el primer paso es descargarse el software de GWT de la página de Google [GWT]. La versión con la que se ha trabajado en este proyecto es la **1.5.3**. Este software descargado permite realizar las siguientes operaciones.

- Utilizar las librerías descargadas para poder compilar la aplicación utilizando nuestra IDE preferido.
- Ejecutar la aplicación en modo hosted el cual es el método de trabajo para poder depurar las aplicaciones. En este modo la aplicación se ejecutará en la máquina virtual Java (JVM) del PC del usuario utilizando un servidor Tomcat incrustado dentro del software descargado.
- Compilar la aplicación y generar el código HTML para distribuir en un servidor de aplicaciones para su visualización en un navegador.

El estudio del producto se centró principalmente en verificar su idoneidad para cumplir las funcionalidades de la aplicación. Este estudio se centró principalmente en los siguientes puntos.

- Estudio de base del aplicativo para verificar la facilidad de uso y el producto resultado.
- Estudio de las capacidades modulares de la aplicación para incluir nuevos componentes que se desarrollaran como parte de la aplicación.
- Estudio de las capacidades de conexión contra sistemas externos a través de servicios Web o llamadas RPC.

Los primeros pasos se centraron en el aprendizaje de esta herramienta mediante el estudio de los manuales descargables desde la propia Web de GWT y de los ejemplos disponibles también en su Web.

Una vez que se dispuso del suficiente conocimiento de la herramienta mediante la implementación de sencillos ejemplos en los que ya se puso de manifiesto su idoneidad para la construcción de aplicaciones web, los siguientes pasos se dedicaron a la construcción de una maqueta de un interfaz que permitiera realizar la mayoría de las operaciones disponibles a través del software hugin.

3.3.2.2 Generación del prototipo

En terminología GWT los componentes de interfaz usuario que están disponibles para su uso en las aplicaciones desarrolladas se denominan widgets. Ejemplos de widgets pueden ser las etiquetas, botones, cuadros de texto e imágenes que podemos ver en una aplicación. Algunos de estos widgets se traducen directamente en elementos HTML que normalmente se usan en la elaboración de una página web, mientras que otros son la composición de muchos elementos HTML combinados con scripts y manejo de eventos.

Los Widgets son uno de los cuatro bloques fundamentales de construcción de aplicaciones GWT (los otros son los paneles, eventos y la comunicación con el servidor).

Todos los Widgets de GWT tienen una alternativa representación DOM que es construida paralelamente con el objeto Java y es con la cual posteriormente se trabaja

En la siguiente ilustración se observa la jerarquía de componentes de GWT.

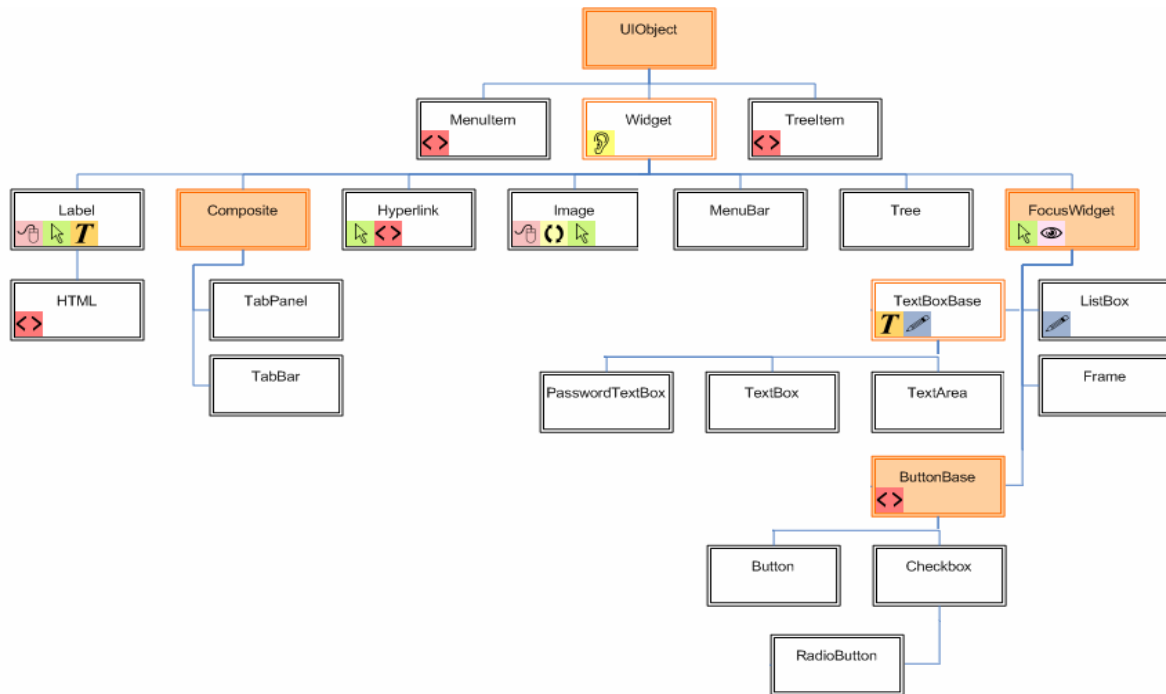


Figura 23: Arquitectura Componentes GWT

La capacidad de orientación a objetos de Java permite que podamos utilizar y extender todos y cada uno de los objetos para construir una aplicación a nuestra medida. Partiendo de este tipo de componentes se comenzó a diseñar el interfaz gráfico.

Tras el análisis de distintas herramientas de composición de mosaicos se puso de manifiesto que disponen de paneles comunes con funcionalidades para cargar, personalizar y visualizar la información final. El boceto de las pantallas de las aplicaciones se empezó a diseñar para que cada una pudiera disponer de un conjunto de funcionalidades específicas.

En la siguiente ilustración se puede ver la estructura de las pantallas realizadas.

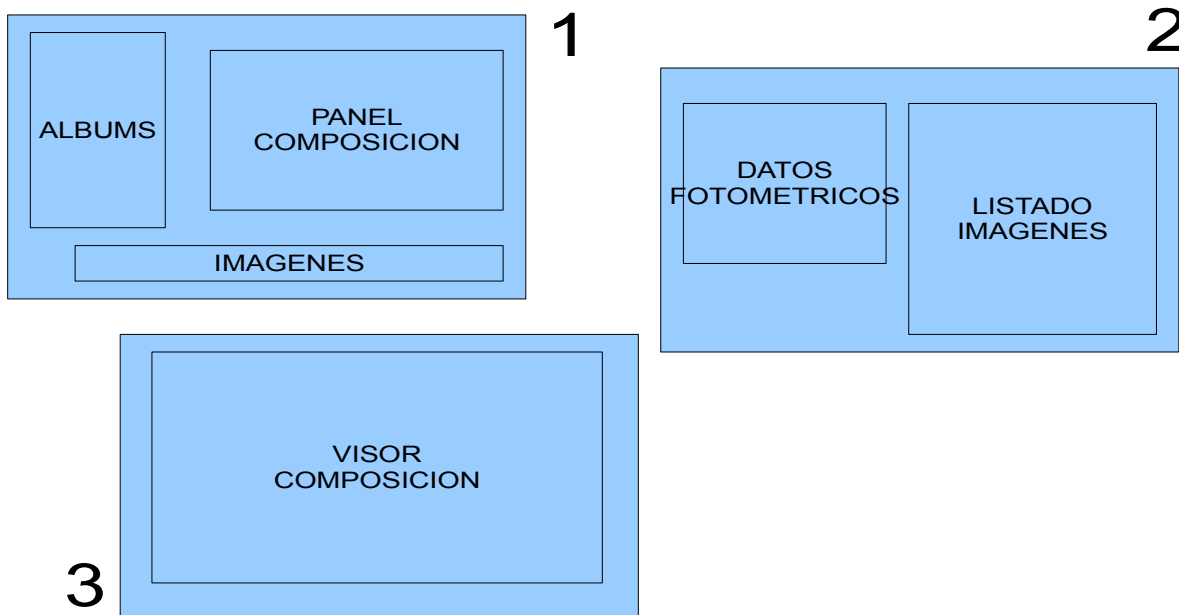


Figura 24: Boceto pantallas

- En el primer panel el usuario va a disponer de una serie de componentes para visualizar los álbumes disponibles en el sistema y las imágenes de cada uno de los mismos. También va a disponer de la posibilidad de poder elegir las imágenes que quiere procesar para generar la composición.
- En el segundo panel el usuario va a poder modificar los valores de composición de las imágenes, modificar valores de distancia focal o exposición con los que se tomaron las imágenes.
- En el tercer panel el usuario puede visualizar el resultado final de la composición así como salvar el resultado a disco.

Para el panel de selección de imágenes se decidió como parte del diseño de este componente diseñar un componente que dispusiera de la capacidad de navegación de imágenes al estilo de los componentes de **flickr** (www.flickr.com) y con soporte **drag & drop** (<http://code.google.com/p/gwt-dn>).

- Mediante la funcionalidad de navegación el usuario puede desplazarse de una forma sencilla por todas las imágenes existentes en un álbum seleccionado mediante la utilización de un desplazamiento suave si el número de imágenes del álbum supera el tamaño de la ventana.
- Mediante el soporte drag&drop el usuario dispone de la posibilidad de una forma sencilla de desplazar imágenes entre distintos paneles. Esto permite que el usuario pueda desplazar solo las imágenes que desea para su procesamiento.

En la siguiente ilustración se puede ver el componente final que se ha diseñado.



Figura 25: Panel SlideShow con soporte Drag & Drop

Este componente se desarrolló con la idea de poder distribuirlo de una forma independiente como uno de los módulos del proyecto de tal forma que podría ser integrado en cualquier componente GWT que se quisiera desarrollar.

3.3.2.3 Capacidades multiidioma

Otro de los puntos importantes en el diseño de la aplicación es el soporte multi-idioma.

Uno de los objetivos de Ciclope es la multiculturalidad del conocimiento y por tanto al ser el generador de mosaicos parte de esta infraestructura la capacidad multiidioma deber ser un factor determinante en su concepción.

GWT permite internacionalización en el uso de Mensajes y Constantes. Para la internacionalización es necesario realizar los siguientes pasos en la aplicación desarrollada.

- Se define un archivo .properties con las claves y los valores
- Seguidamente se define una interfaz que herede de Constants o Messages. Se utiliza Constants cuando los valores a obtener son estáticos y Messages cuando en el valor hay que incluir algún parámetro (contenido dinámico). El nombre de la interfaz debe ser el mismo que el del archivo .properties.
- La interfaz define los métodos de acceso a los valores. Los nombres de los métodos deben casar con los nombres de las keys en el .properties.
- GWT genera una implementación de la interfaz para acceder a los properties.
- La implementación generada depende del locale del cliente que se indica en la llamada a la aplicación.

La forma de poder utilizar las aplicaciones en diferentes idiomas se basa en uso del parámetro locale.

<http://<IP>:/MosaicImageComposer/?locale=en> (Acceso Ingles)

<http://<IP>:/MosaicImageComposer/?locale=es> (Acceso Castellano)

3.3.3 Integración cliente servidor

Mientras se fue desarrollando la aplicación de cliente se comenzó a diseñar el módulo servidor para ir verificando las funcionalidades que se iban desarrollando. Para esta integración se realizó un estudio

por una parte del modelo de integración cliente/servidor así como los mecanismos necesarios para llevar a cabo las operaciones en el servidor.

Para la integración cliente/servidor se analizó las capacidades de la tecnología GWT para poder acceder a los servicios ofrecidos por el servidor.

3.3.3.1 Llamadas a procedimiento remoto

El mecanismo para interactuar con el servidor a través de la red es el RPC “Remote Procedure Call”. El RPC en Google Web Toolkit permite fácilmente al cliente enviar y recibir objetos Java sobre HTTP. En la Figura 26 se puede ver la arquitectura RPC de GWT.

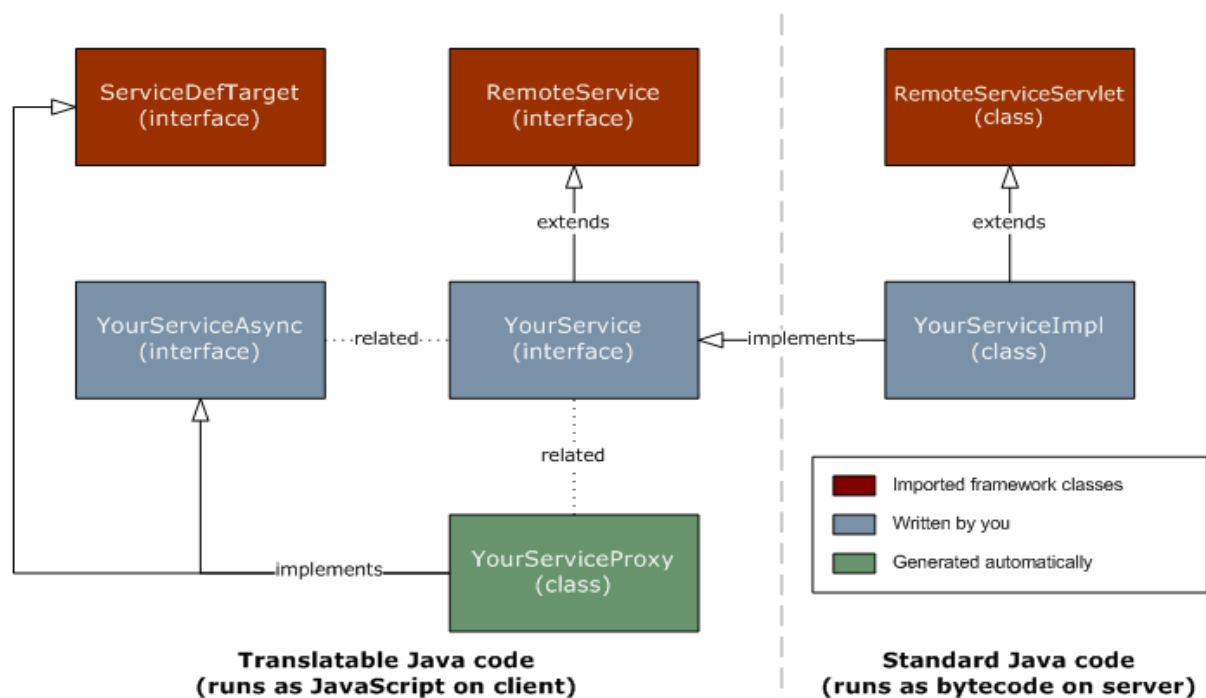


Figura 26: Arquitectura RPC en GWT

Una vez desarrollado la interfaz del cliente se comenzó a trabajar en el módulo servidor para poder completar la integración.

3.3.3.2 Arquitectura servidor

Para el diseño de la aplicación en el servidor se ha decidido utilizar el patrón modelo-vista-controlador (MVC). MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Aunque el modelo de aplicación de GWT no sigue el esquema de una aplicación MVC típica en la cual se realiza

una recarga de la página del cliente en cada petición, es importante separar en la aplicación.

Para este tipo de patrón existen diferentes tecnologías que pueden ser las adecuadas para cada uno de los componentes.

- **Vista.** Para la vista existen distintos frameworks: los más importantes en el mercado son Struts, Struts2, JSF y Spring-MVC.
- **Lógica de Negocio:** El más usado y el más recomendable en sin duda Spring. En este trabajo se ha optado por utilizar Spring.
- **Acceso a Datos:** Con respecto a esta parte del MVC lo más común es trabajar con Hibernate pero si estamos trabajando con una aplicación que ya está diseñada y tiene un modelo de Base de Datos ya diseñado la opción de Ibatis empieza a despertar interés. Cuando el modelo de Base de Datos de la aplicación tampoco es muy complicado Ibatis suele ser una buena elección. En nuestro caso se ha optado por Ibatis para la capa de acceso a Datos.

Para desarrollar una nueva interfaz de servicio, se comienza por crear una interfaz que herede de **RemoteService** donde se definen los métodos que se van a implementar y que van a dotar al sistema de la capacidad de generar mosaicos.

```
public interface MosaicoService extends RemoteService{

    boolean initMosaic();

    boolean testMosaic();

    //Metodos para la generacion del panorama

    ResultUploadProcess uploadImages(ArrayList<ImagenMosaico> images,List<ConfigMosaico> config);

    ResultAlignProcess alignImages(ArrayList<ImagenMosaico> images,List<ConfigMosaico> config);

    ResultAlignProcess getStatusAlign(int idProcess);

    ResultCreatePanoramaProcess createPanorama(int idProcess,List<ConfigMosaico> config);

    ResultCreatePanoramaProcess getStatusCreatePanorama(int idProcess);

}
```

Cualquier implementación de este servicio en el lado del servidor hereda a **RemoteServiceServlet** e implementa esta interfaz de servicio definida anteriormente.

Cada servicio finalmente necesita realizar algunos procesos con el fin de responder peticiones a los clientes. Cada proceso del lado del servidor ocurre en la implementación del servicio, que está basada en la conocida arquitectura de los Servlets.

```
public class MosaicoServiceImpl implements MosaicoService,MosaicoCatalogService,MosaicoConfigService{

    final String beanResource="com/ciclope/mosaicos/resources/beans/beans.xml";

    ApplicationContext applicationContext = new ClassPathXmlApplicationContext(beanResource);

    MosaicoManager mosaicoManagerApplicationContext = (MosaicoManager)applicationContext.getBean("mosaicoManager");

    public MosaicoServiceImpl() {
```

```

    }

    public boolean initMosaic(){

        return mosaicoManagerApplicationContext.initMosaic();

    }

    public ResultUploadProcess uploadImages(ArrayList<ImagenMosaico> images,List<ConfigMosaico> config) {

        return mosaicoManagerApplicationContext.uploadImages(images,config);

    }

    .....

```

Antes de que se pueda realiza una invocación remota desde el cliente, se crea otra interfaz que en GWT se denomina asíncrona basada en la interfaz de servicio original.

```

public interface MosaicoServiceAsync {

    void initMosaic(AsyncCallback callback);

    void testMosaic(AsyncCallback callback);

    void uploadImages(ArrayList<ImagenMosaico> images,List<ConfigMosaico> config,AsyncCallback callback);

    void alignImages(ArrayList<ImagenMosaico> images,List<ConfigMosaico> config,AsyncCallback callback);

    void getStatusAlign(int idProcess,AsyncCallback callback);

    void createPanorama(int idProcess,List<ConfigMosaico> config,AsyncCallback callback);

    void getStatusCreatePanorama(int idProcess,AsyncCallback callback);

}

```

La forma de invocar métodos asíncronos requiere que quien realice la invocación pase un objeto de tipo “**callback**” que pueda ser notificado cuando la llamada asíncrona esté completa, y el servidor haya devuelto la respuesta, ya que quien realice la invocación no puede bloquearse hasta que la llamada esté completa. Cuando se retorna el resultado se invoca el método en el objeto callback.

```

AsyncCallback<Object> serviceCallback = new AsyncCallback<Object>() {

    public void onSuccess(Object result) {

        //Resultado correcto invccacion

    }

    public void onFailure(Throwable caught) {

        //Resultado incorrecto invccacion

    }

};

```

Por la misma razón, los métodos asíncronos no tienen tipo de retorno; siempre deben retornar void

(Este es uno de las mayores problemas que tienen los desarrolladores a la hora de definir este tipo de métodos y que provocan problemas en tiempo de ejecución).

Después que una llamada asíncrona es hecha, la información de retorno a quien hizo la invocación es hecha a través de un objeto callback que dispone de dos métodos (onSuccess y onFailure) para tratar el caso de éxito y el caso de error en la invocación del servicio.

Para invocar el servicio simplemente tenemos que realizar los siguientes pasos.

1. Instanciar la interfaz de servicio usando GWT.create().
2. Especificar un URL para el servicio usando ServiceDefTarget.
3. Crear un objeto callback asíncrono para ser notificado cuando el RPC se haya completado.
4. Realizar la llamada/invocación.

```
(1) MosaicoServiceAsync service = (MosaicoServiceAsync)GWT.create(MosaicoService.class);

ServiceDefTarget endpoint = (ServiceDefTarget) service;

(2) endpoint.setServiceEntryPoint(GWT.getModuleBaseURL() + "/MosaicoService");

(3) AsyncCallback<Object> serviceCallback = new AsyncCallback<Object>() {

    public void onSuccess(Object result) {

        //.....

    }

    public void onFailure(Throwable caught) {

        //.....

    }

};

(4) service.alignImages(images,model.getConfig(),serviceCallback);
```

3.3.3.3 Lógica de negocio y Acceso a Datos

Una vez que las peticiones llegan al servidor es preciso realizar el tratamiento de las mismas y es ahí donde entran los framework Spring e Ibatis.

El modelo de spring está basado en la definición en una serie de ficheros XML de los objetos (beans) que van a implementar la lógica de negocio de la aplicación. Spring se encarga de instanciar estos objetos en lugar de que sea el usuario el encargado de dicha instanciación mediante el uso de inyección de dependencias y otros mecanismos accesorios.

En este proyecto se han diseñado e implementado cuatro DAO (Data Access Object) para llevar a cabo las funcionalidades definidas en el proyecto.

- **AlbumMosaicoDAO.** Encargado de la gestión de los álbumes disponibles en el sistema.
- **ConfigMosaicoDAO.** Encargado de obtener los parámetros necesarios para el correcto funcionamiento de la aplicación.
- **ImageMosaicoDAO.** Encargado de gestionar las operaciones de composición de imágenes
- **UserMosaicoDAO.** Encargado de la gestión de usuarios.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>

  <bean
class="com.ciclope.mosaicos.server.business.manager.implementations.MosaicoManagerImpl"
name="mosaicoManager">

    <property name="configMosaicoDAO"><ref bean="configMosaicoDAO"/></property>

    <property name="imagenMosaicoDAO"><ref bean="imagenMosaicoDAO"/></property>

    <property name="albumMosaicoDAO"><ref bean="albumMosaicoDAO"/></property>

    <property name="userMosaicoDAO"><ref bean="userMosaicoDAO"/></property>

  </bean>

  <bean id="userMosaicoDAO" class=" com.ciclope.mosaicos.server.business.dao.implementations.UserMosaicoDAOImpl">

    <property name="sqlMapClient"><ref bean="sqlMapClient"/></property>

  </bean>

  .....

  .....

  .....
```

3.3.4 Visor de imágenes

El tamaño de las panorámicas generadas puede ser mucho mayor que el ancho de la ventana del navegador por lo que para ver el resultado final de la composición y poder observar los detalles de la misma, el usuario va a tener la posibilidad de poder descargar la imagen a su PC y trabajar con ellas con algún programa específico o como alternativa va a disponer de la posibilidad de poder navegar y realizar operaciones de zoom sobre la propia imagen generada directamente en la aplicación.

Para poder realizar esta operación se realizó un estudio de varias alternativas opensource para poder visualizar la información con la posibilidad de que el usuario pudiera desplazarse por la imagen y realizar sencillas operaciones para alejar y acercar la imagen.

Lo que se buscó principalmente en estas alternativas fue la capacidad de poder integrar la herramienta dentro del sistema de composición como una pestaña adicional.

Entre los productos que se analizaron están los siguientes:

- IIPMooViewer (<http://iipimage.sourceforge.net/>)
- Microsot SilverLight (DeepZoom)
- Yawah (<http://www.yawah.com/>)
- Zoomify (<http://www.zoomify.com/>)

De entre estos productos el más interesante por su sencillez y por el resultado final y debido a que no necesita licencia para su uso estaba el **IIPMooViewer**.

IIPMooViewer básicamente es un servidor de imágenes con las siguientes características.

- Software especializado en servir imágenes. Arquitectura cliente-servidor e interacción del cliente a través API's públicas
- Permite el almacenamiento de una imagen (formato base) y generación dinámica de x formatos (tamaños/resolución).
- Dispone de herramientas para navegación en tiempo real por imágenes de tamaño mega y gigapixel. En este caso dispone de visores, ajax, flash o applet. En nuestro caso hemos utilizado el visor AJAX.
- Optimiza el tráfico cliente - servidor con respecto al ancho de banda y gestiona con gran eficiencia la memoria del servidor

3.3.4.1 IIP

El software está basado en el protocolo **IIP** (Internet Image Protocol) formato definido y mantenido por el Digital Imaging Group (DIG).

- La comunicación con el servidor está basada en el protocolo Internet Imaging Protocol (IIP) el cual está basado en HTTP.
- Trabaja sobre las múltiples posibilidades de “multiresolution tiled images”
- Trafico HTTP de secciones de la imagen a una determinada resolución
- Las secciones son servidas y comprimidas dinámicamente en formato JPEG

3.3.4.2 Formato Piramidal

Es el formato de referencia para trabajar con el protocolo IIP y es el que utiliza el IIPMooViewer.

Es un formato multi-resolución, lo que significa que contiene datos de la imagen para múltiples resoluciones y ofrecen la posibilidad de leer porciones de la imagen en distintos niveles de resolución y sobre todo a gran velocidad.

Las imágenes se almacenan en secciones rectangulares del mismo tamaño. Ancho y alto múltiplo de 16 píxeles (las especificaciones IIP limitan a 64×64). Las *tiles* (teselas) son comprimidas individualmente. Dispone de varios formatos de compresión: PackBits, Huffman, LZW, JPEG.

Para la generación del formato piramidal a partir de las imágenes generadas por el hugin se ha trabajado con varias herramientas ya que las imágenes que genera hugin no disponen de la capacidad de multi-resolución.

4 Diseño de la aplicación

En este apartado se definen los requisitos que debe cumplir el aplicativo a desarrollar, según los requerimientos de los usuarios, los casos de uso así como el modelo de datos de la aplicación.

4.1 Requisitos de usuario

A cada requisito se ha asociado un código, un nombre y una descripción en lenguaje coloquial, buscando la mayor facilidad de lectura posible. Esta identificación servirá para referir cualquier desarrollo posterior del proyecto.

Un requisito representa un objetivo del sistema que se desea alcanzar, y que proporcionará un beneficio a los usuarios.

En este apartado se especifican los requisitos que se deberán satisfacer con este proyecto:

- Requisitos Funcionales, que hacen referencia a las funcionalidades que deberán ser incluidas en el sistema.
- Requisitos No Funcionales, que hacen referencia a las funcionalidades que deberán ser soportadas en el sistema.
 - Requisitos de Desarrollo, que detalla los procedimientos y metodologías que se seguirán durante este proyecto.
 - Requisitos Tecnológicos, relacionadas con las tecnologías que se incluirán.
 - Requisitos de Seguridad, necesarios para mantener la coherencia y seguridad en la herramienta.
 - Requisitos de documentación, a generar durante la ejecución del proyecto.
 - Requisitos de implantación del sistema .

En la Figura 27 se muestran los requisitos formales del sistema.

4.1.1 Requerimientos Formales

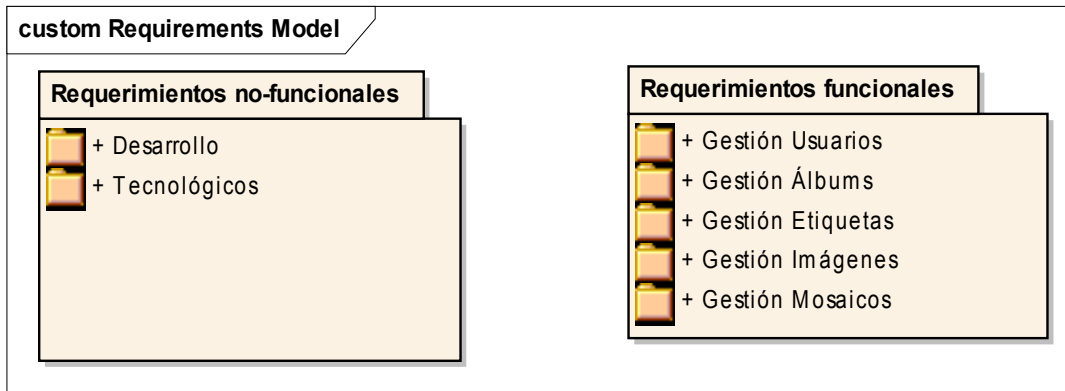


Figura 27: Requisitos del Sistema

4.1.1.1 Requerimientos funcionales

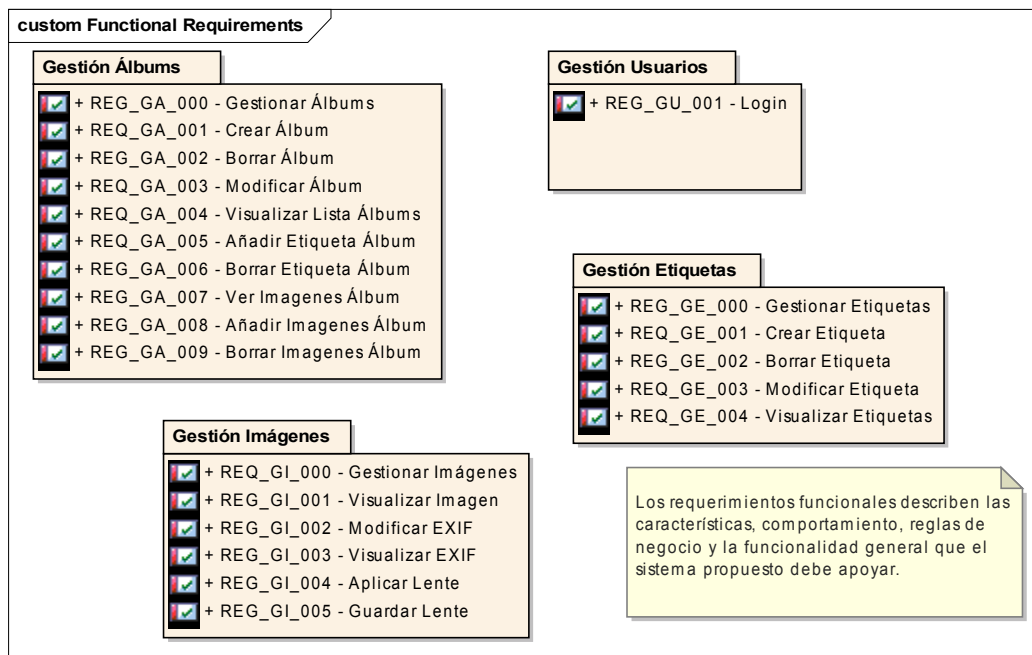


Figura 28: Requerimientos funcionales

Note

Estado: Proposed

Versión: 1.0

Prioridad:

Los requerimientos funcionales describen las características, comportamiento, reglas de negocio y la funcionalidad general que el sistema propuesto debe apoyar.

4.1.1.1.1 Gestión Usuarios

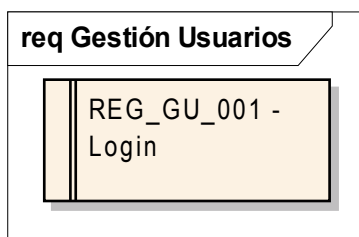


Figura 29: Gestión de usuarios

REG_GU_001 - Login			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario deberá realizar Login en la aplicación antes de poder utilizar sus funcionalidades.		

4.1.1.1.2 Gestión Álbumes

En la Figura 30 se muestran los requisitos de usuario para la gestión de álbumes.

REG_GA_000 - Gestionar Álbumes			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema debe disponer de la posibilidad de poder realizar una gestión integral de álbumes, permitiendo crear, borrar y modificar los álbumes del catálogo.		

REQ_GA_001 - Crear Álbum			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema deberá permitir que el usuario puede crear álbumes con el objetivo de poder incluir imágenes dentro de los mismos		

REG_GA_002 - Borrar Álbum			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario de la aplicación tendrá la capacidad de borrar álbumes del catálogo disponible en el sistema		

REQ_GA_003 - Modificar Álbum			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario de la aplicación tendrá la capacidad de modificar el nombre de los álbumes del catálogo disponible en el sistema		

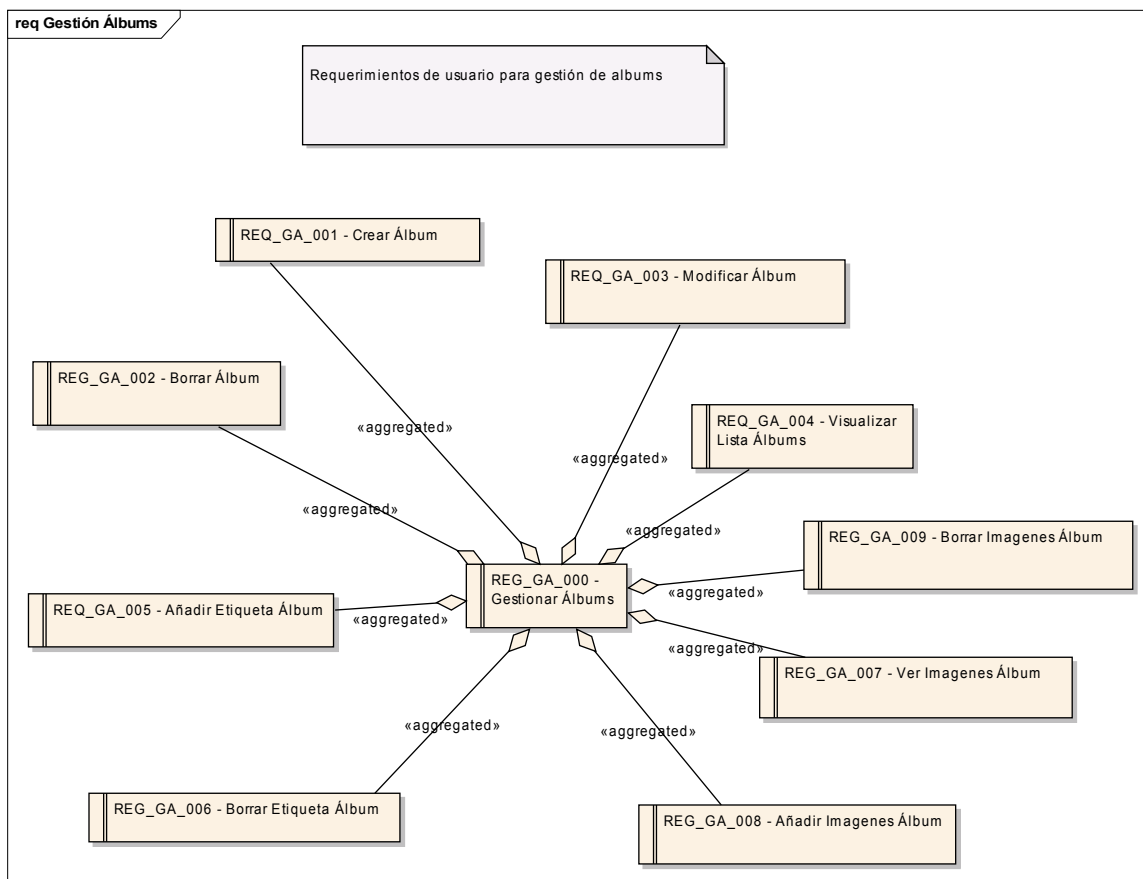


Figura 30: Gestión del álbumes

REQ_GA_004 - Visualizar Lista Álbumes

«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario tendrá la posibilidad de poder visualizar los álbumes disponibles en el sistema		

REQ_GA_005 - Añadir Etiqueta Álbum

«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá añadir nuevas etiquetas a un álbum permitiendo que pueda ser catalogado bajo distintas puntos de vista.		

REG_GA_006 - Borrar Etiqueta Álbum

«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá borrar etiquetas de un álbum.		

REG_GA_007 - Ver Imágenes Álbum

«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá visualizar las imágenes que tiene un álbum		

REG_GA_008 - Añadir Imágenes Álbum

«Functional»	<u>Estado:</u> Proposed	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá incluir imágenes en el álbum desde su PC. El sistema subirá las imágenes al servidor para su proceso.		

REG_GA_009 - Borrar Imágenes Álbum

«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá borrar las imágenes que tiene un álbum de forma individual.		

4.1.1.1.3 Gestión Etiquetas

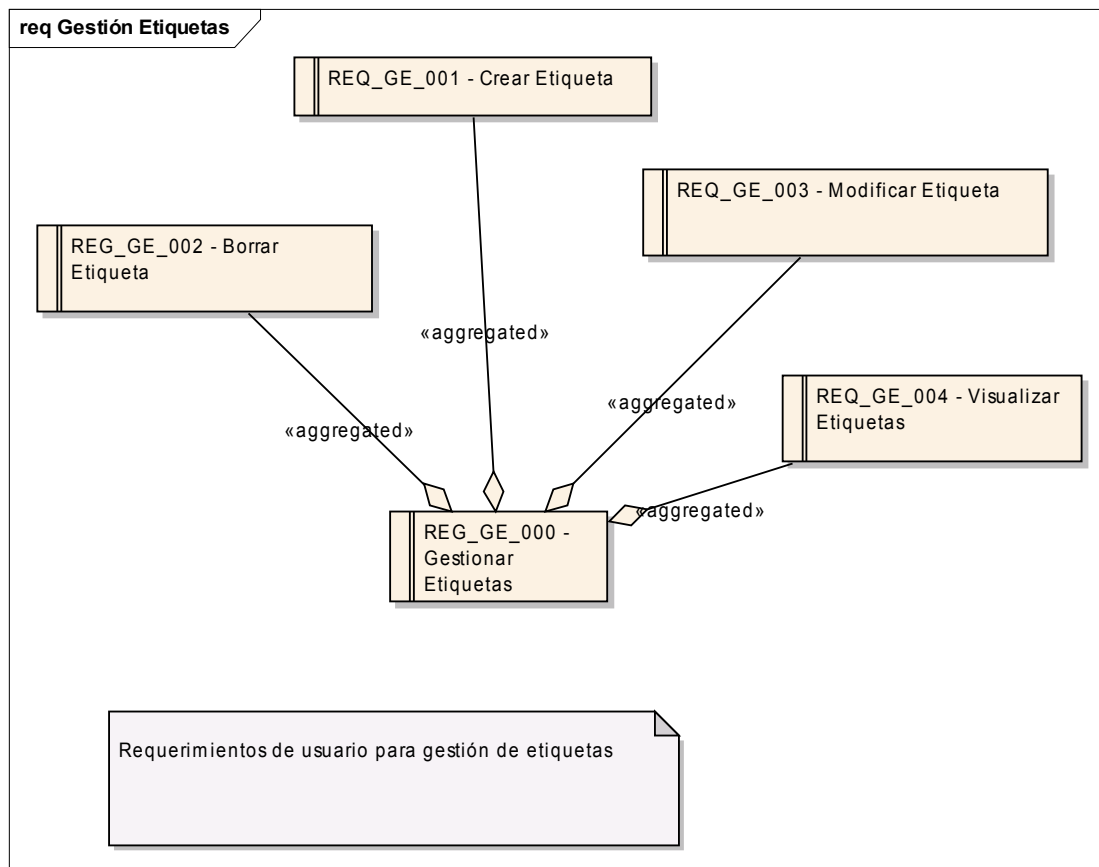


Figura 31: Gestión de etiquetas

REG_GE_000 - Gestionar Etiquetas			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema debe disponer de la posibilidad de poder realizar una gestión integral de etiquetas, permitiendo crear, borrar y modificar las etiquetas disponibles		

REQ_GE_001 - Crear Etiqueta			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema deberá permitir que el usuario puede crear etiquetas para poder asignárselas a los álbumes		

REG_GE_002 - Borrar Etiqueta			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema deberá permitir que el usuario puede borrar etiquetas.		

REQ_GE_003 - Modificar Etiqueta			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema deberá permitir que el usuario puede modificar el nombre de las etiquetas para poder asignárselas a los álbumes		

REQ_GE_004 - Visualizar Etiquetas			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario tendrá la posibilidad de poder visualizar las etiquetas disponibles en el sistema		

4.1.1.1.4 Gestión Imágenes

En la Figura 32 se pueden observar los requisitos de usuario para la gestión de imágenes.

REQ_GI_000 - Gestionar Imágenes			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema debe disponer de la posibilidad de poder realizar una gestión integral de imágenes, permitiendo visualizar su información así como modificar los parámetros de exif para el proceso de generación del mosaico		

REG_GI_001 - Visualizar Imagen			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá visualizar las imágenes para ver sus características		

REG_GI_002 - Modificar EXIF			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá modificar la información de metadatos de la imagen.		

REG_GI_003 - Visualizar EXIF			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá ver la información de características de imágenes previa al proceso de alineación		

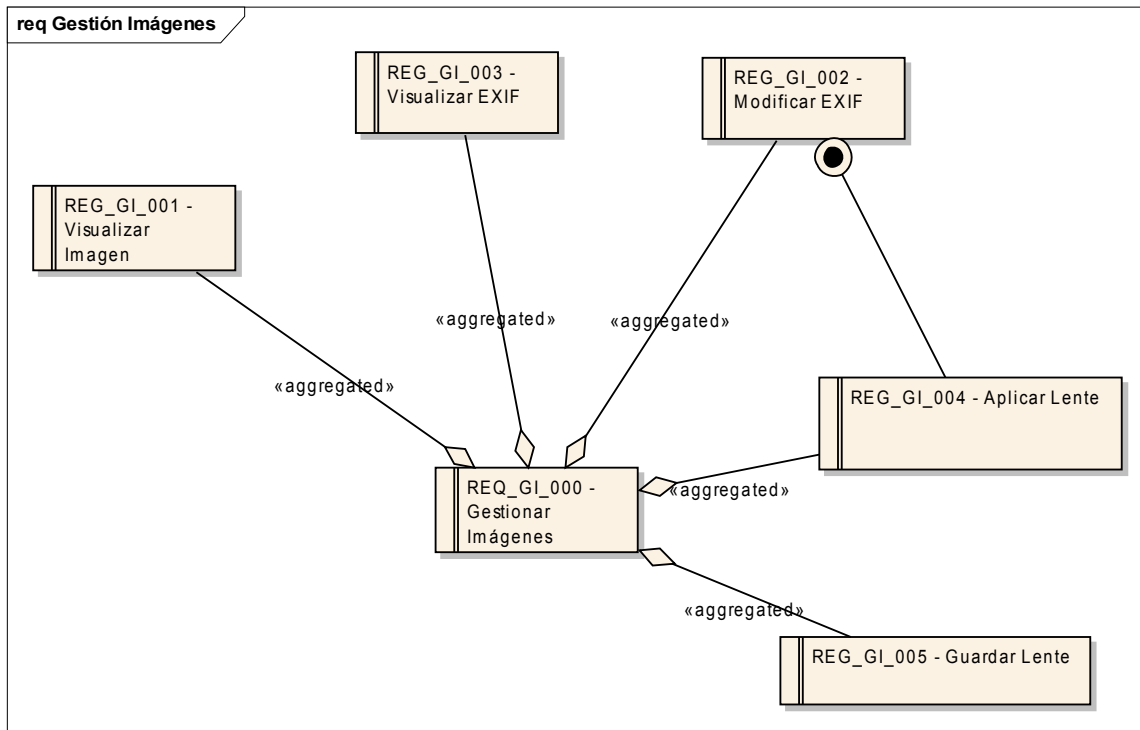


Figura 32: Gestión de imágenes

REG_GI_004 - Aplicar Lente			
«Functional»	<u>Estado</u> : Implemented	<u>Versión</u> : 1.0	<u>Prioridad</u> : Medium
	El usuario podrá aplicar una lente a una imagen con el objetivo de actualizar todos los valores almacenados en la lente seleccionada. Los valores que se actualizan son los valores de EXIF almacenados		

REG_GI_005 - Guardar Lente			
«Functional»	<u>Estado</u> : Proposed	<u>Versión</u> : 1.0	<u>Prioridad</u> : Medium
	El usuario tendrá la posibilidad de poder guardar una combinación de características con un nombre de lente definido.		

4.1.1.1.5 Gestión Mosaicos

En la Figura 33 se puede ver el diagrama de requisitos de la gestión de mosaicos.

REG_GM_000 - Gestionar Mosaicos			
«Functional»	<u>Estado:</u> Proposed	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema debe disponer de la posibilidad de poder realizar una gestión integral de los mosaicos generados por los usuarios.		

REQ_GM_001 - Crear Mosaico			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema deberá permitir que el usuario puede crear un mosaico con las imágenes contenidas en un álbum		

REQ_GM_002 - Visualizar Mosaico			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario tendrá la posibilidad de poder visualizar el mosaico generado con las imágenes disponibles en el álbum		

REQ_GM_003 - Personalizar Mosaico			
«Functional»	<u>Estado:</u> Proposed	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá modificar los parámetros que afectan al proceso de generación de un mosaico		

REQ_GM_004 - Pregenerar Mosaico			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema deberá permitir que el usuario puede crear un mosaico con las imágenes contenidas en un álbum La diferencia con la generación del mosaico es que la calidad del mosaico es inferior al proceso final. En esta fase se debe de realizar un proceso de alineación de las imágenes para comprobar que tienen puntos comunes		

REG_GM_005 - Añadir Imagen a Mosaico			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario tendrá la posibilidad de añadir imágenes al mosaico		

REG_GM_006 - Borrar Imagen de Mosaico			
«Functional»	<u>Estado:</u> Proposed	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario tendrá la posibilidad de borrar imágenes del mosaico		

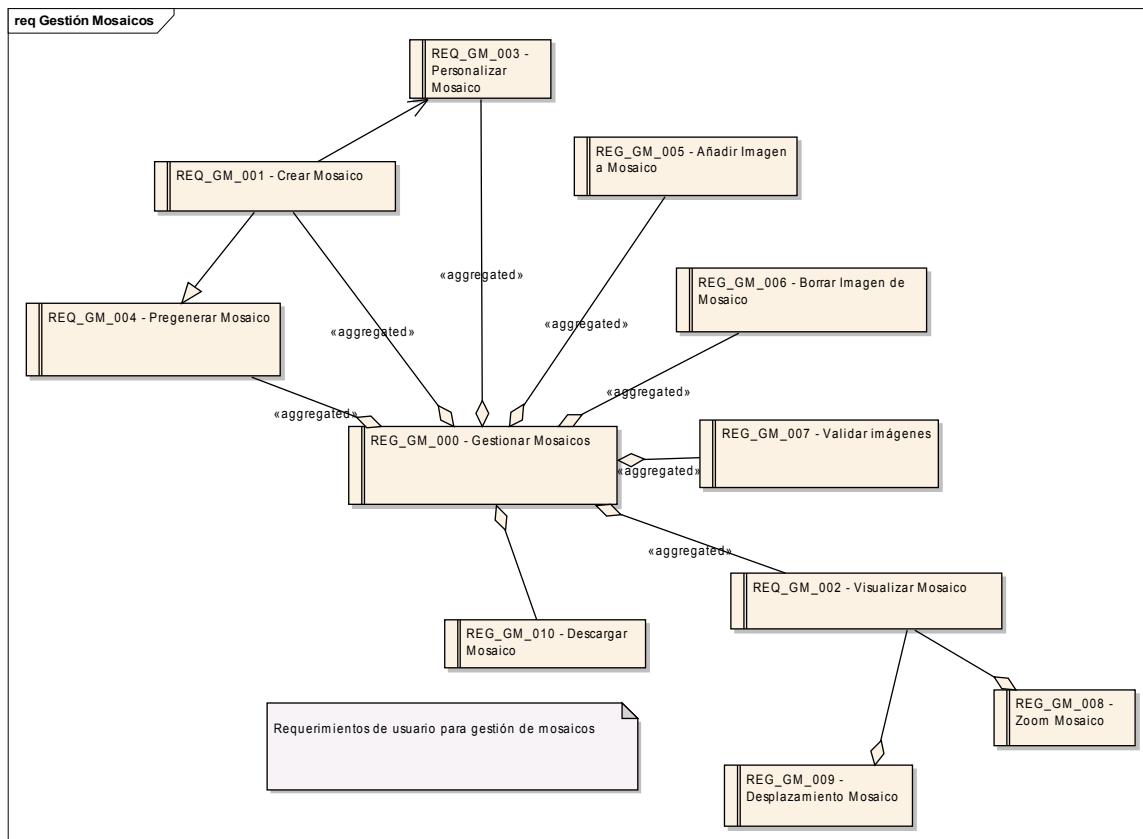


Figura 33: Gestión de mosaicos

REG_GM_007 - Validar imágenes

«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá enviar las imágenes al servidor para la validación de las mismas.		

REG_GM_008 - Zoom Mosaico

«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario tendrá la posibilidad realizar zoom más y menos sobre la imagen generada.		

REG_GM_009 - Desplazamiento Mosaico

«Functional»	<u>Estado:</u> Proposed	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
--------------	-------------------------	---------------------	--------------------------

	El usuario tendrá la posibilidad de desplazarse sobre la imagen generada
--	--

REG_GM_010 - Descargar Mosaico			
«Functional»	<u>Estado:</u> Implemented	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El usuario podrá descargar la imágenes generada en su PC		

4.1.1.2 Requerimientos no-funcionales

En la Figura 34 se pueden ver los requisitos no funcionales del proyecto.

Estado: Proposed Versión: 1.0 Prioridad:
 Estos paquetes contienen los requisitos no funcionales especificados para el nuevo sistema. Estos suelen describir los criterios de rendimiento, fiabilidad, seguridad y otros parámetros de funcionamiento.

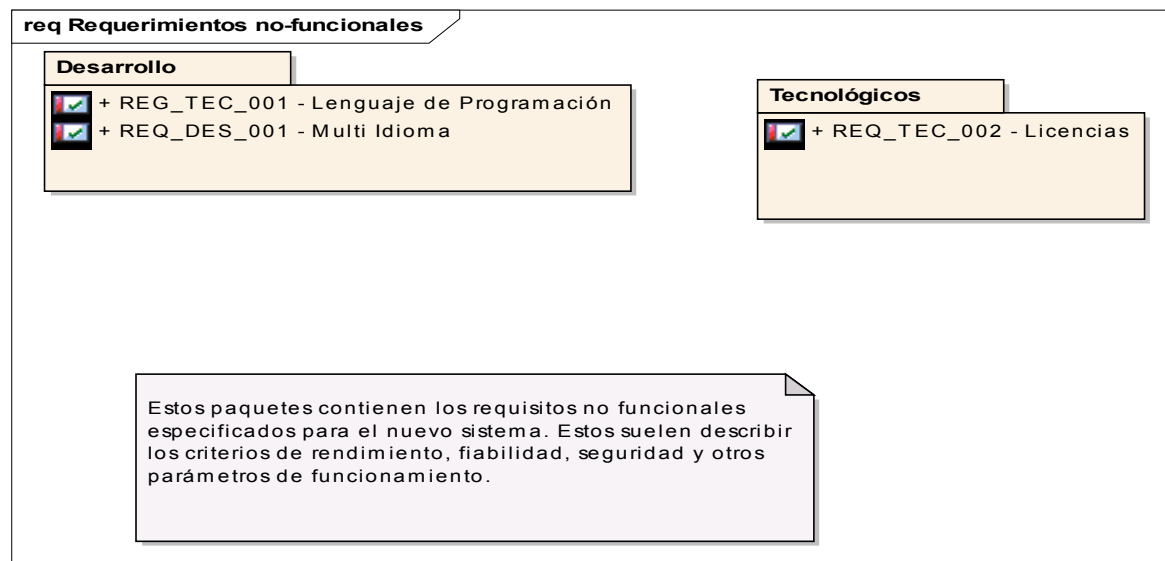
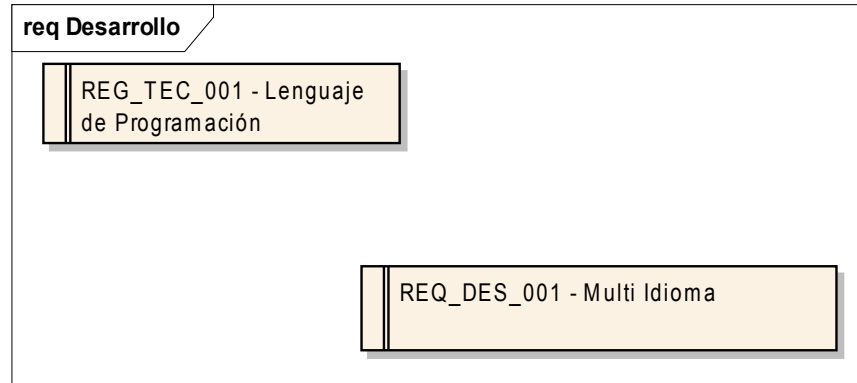


Figura 34: Requerimientos no funcionales

4.1.1.2.1 Desarrollo



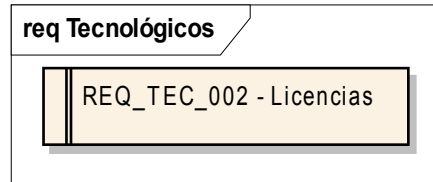
REG_TEC_001 - Lenguaje de Programación

«Functional»	<u>Estado</u> : Proposed	<u>Versión</u> : 1.0	<u>Prioridad</u> : Medium
	Se utilizará el lenguaje de programación Java (GWT) para el desarrollo del módulo cliente.		

REQ_DES_001 - Multi Idioma

«Functional»	<u>Estado</u> : Approved	<u>Versión</u> : 1.0	<u>Prioridad</u> : Medium
	El sistema, en cuanto a sus módulos, aplicaciones y herramientas se hará configurable para permitir que pueda ser visualizado en varios idiomas. Para ello se colocarán todos los ítems de menús, opciones, interfaces, ayudas y demás elementos en ficheros o tablas aparte para facilitar su traducción. La opción de trabajar en una u otra lengua será uno de los elementos de configuración básica del sistema seleccionables según se vayan implementando las traducciones.		

4.1.1.2.2 Tecnológicos



REQ_TEC_002 - Licencias			
«Functional»	<u>Estado:</u> Proposed	<u>Versión:</u> 1.0	<u>Prioridad:</u> Medium
	El sistema se desarrollará mediante el uso de productos open-source sin que sea necesario una licencia para su uso		

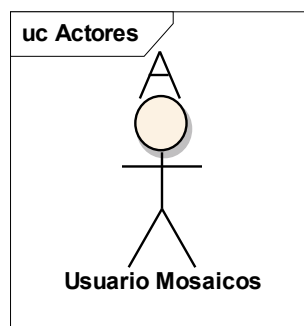
4.2 Modelo de casos de uso

Se partirá del catálogo de requisitos para detallar las funcionalidades del sistema, utilizando la notación UML para detallar los casos de uso donde se considere necesario.

4.2.1 Diagrama de casos de uso

En la Figura 35 se muestran los diagramas de caso de uso del sistema.

4.2.2 Descripción de los actores



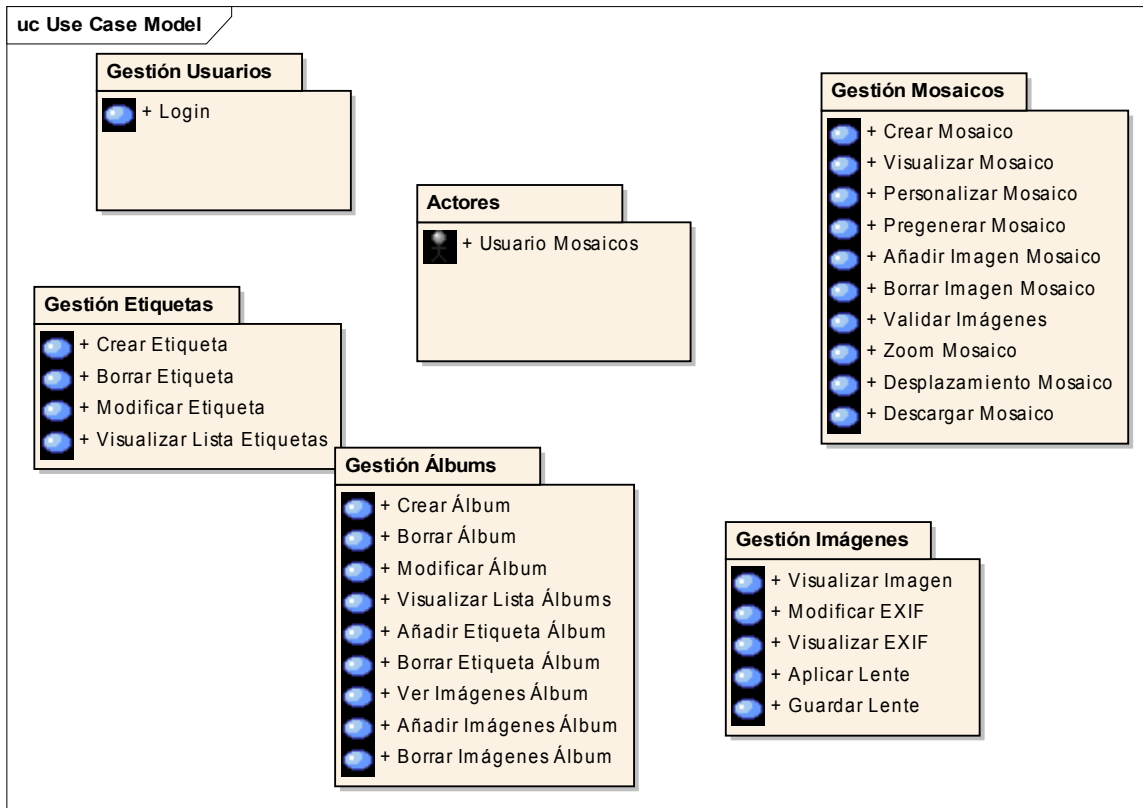


Figura 35: Diagrama de casos de uso

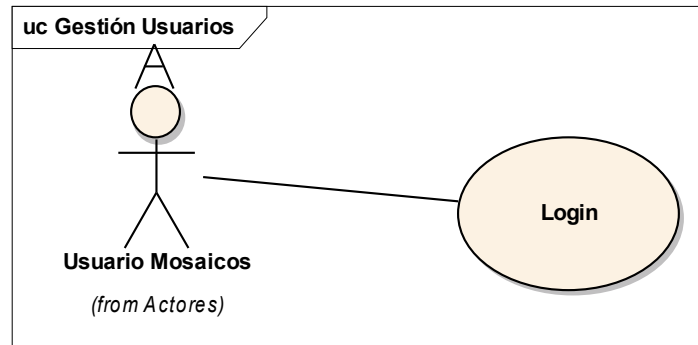
1. Usuario Mosaicos

Usuario que realiza el proceso de generación de mosaicos utilizando los recursos proporcionados por la aplicación.

Representa cualquier usuario del sistema Ciclope Astro que tiene acceso al módulo de generación de imágenes panorámicas.

4.2.3 Casos de uso en formato breve

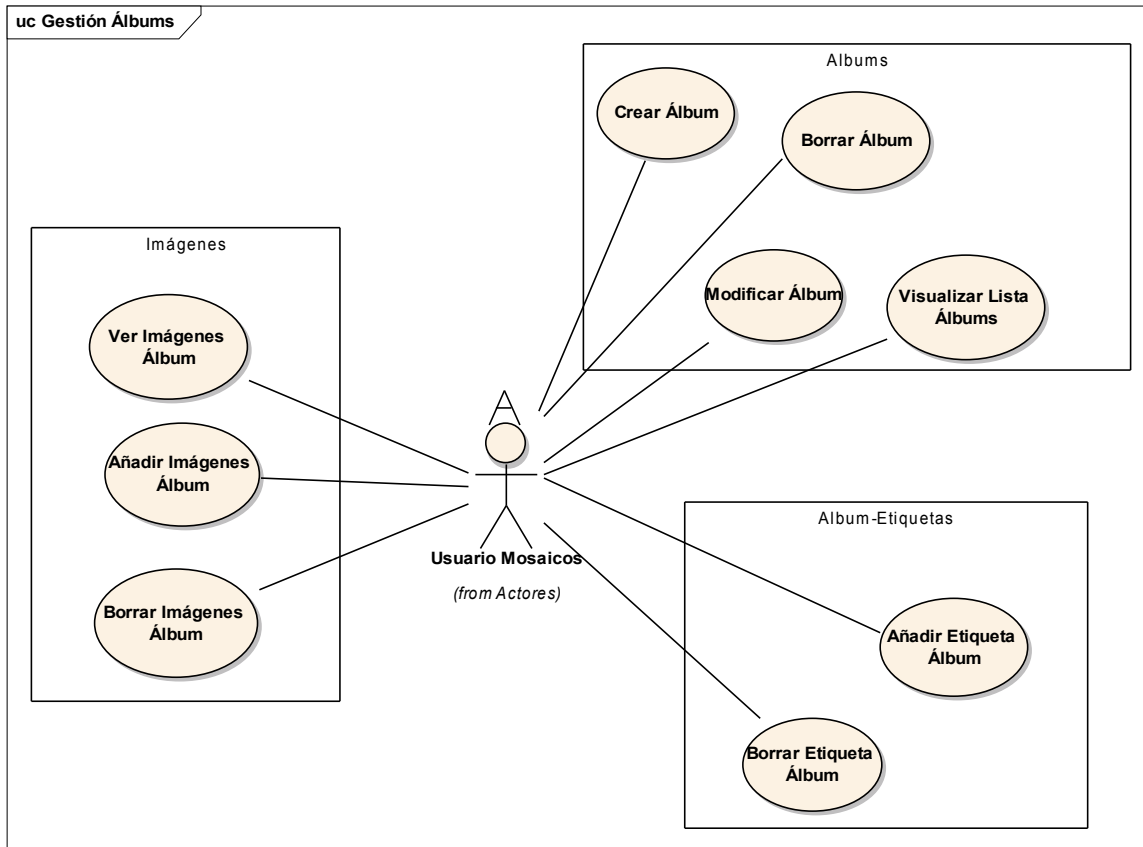
4.2.3.1 Gestión Usuarios



2.Login

<u>Flujo de Eventos</u>	
<i>Flujo Básico</i>	
1.El usuario introduce el login y el password. 2.Se valida contra el sistema Ciclope para verificar la información. 3.Si los datos son correctos el sistema muestra la pantalla de bienvenida de la aplicación.	
<i>Flujo alternativo</i>	
1.Si los datos no son correctos se informa al usuario del error de autenticación 2.Se vuelve a mostrar la pantalla de usuario y password	

4.2.3.2 Gestión Álbumes



3.Crear Álbum

El usuario dispone de la posibilidad de poder crear álbumes sobre los que incluir nuevas imágenes.

Flujo de Eventos

Flujo básico

- 1.El usuario se sitúa sobre el nodo catálogo.
- 2.Una vez seleccionado el nodo sobre el menú contextual que le aparece al usuario selecciona la opción de creación de álbumes.
- 3.El sistema automáticamente coloca al final del árbol el álbum generado. El nombre del álbum que se genera tiene un nombre aleatorio que posteriormente el usuario puede cambiar
- 4.El usuario visualmente puede ver el álbum creado

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber seleccionado el nodo de catálogo.	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

4. *Borrar Álbum*

El usuario dispone de la posibilidad de poder borrar álbumes.

<u>Flujo de Eventos</u>
<i>Flujo básico</i>
<ol style="list-style-type: none"> 1.El usuario se sitúa sobre el nodo a borrar. 2.Una vez seleccionado el nodo sobre el menú contextual que le aparece al usuario selecciona la opción de borrar el álbum seleccionado. 3.El sistema borra el álbum y refresca la lista de álbumes del usuario 4.El usuario visualmente puede ver el álbum borrado

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber selecciona un álbum de los disponibles en el sistema	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

5. *Modificar Álbum*

El usuario dispone de la posibilidad de poder modificar el nombre del álbum.

<u>Flujo de Eventos</u>
<i>Flujo básico</i>
<ol style="list-style-type: none"> 1.El usuario se sitúa sobre el nodo a modificar. 2.Una vez seleccionado el nodo sobre el menú contextual que le aparece al usuario este selecciona la opción de modificar el álbum seleccionado. 3.El sistema presenta al usuario un cuadro de texto sobre el nodo a modificar para que el

usuario puede introducir el nuevo nombre del álbum
 4.El sistema modifica el nombre del álbum en la Base de Datos
 5.El usuario visualmente puede ver el nuevo nombre del álbum

Flujo alternativo

1.Si el nombre del álbum ya existe el sistema vuelve a colocar el cursor sobre el álbum hasta que el usuario decida seleccionar un nuevo nombre.
 2.Si desea cancelar pulsando el botón de escape el nombre del álbum se deja con el nombre original.

Condiciones

El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
---	----------------------

El usuario debe haber selecciona un álbum de los disponibles en el sistema	<u>Pre-Condición</u>
--	----------------------

La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>
---	-----------------------

6. *Visualizar Lista Álbumes*

El usuario dispone de la posibilidad de poder visualizar la lista de álbumes disponibles en el sistema

Flujo de Eventos

Flujo básico

1.El usuario dispone de un enlace para refrescar la lista de álbumes disponibles en el sistema
 2.Una vez seleccionado ese enlace el sistema accede a la Base de Datos para obtener la lista de álbumes del usuario.

Condiciones

El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
---	----------------------

7. *Añadir Etiqueta Álbum*

El usuario dispone de la posibilidad de poder añadir etiquetas a los álbumes.

Flujo de Eventos

Flujo básico

- 1.El usuario selecciona el álbum sobre el que quiere añadir la etiqueta.
- 2.El usuario sobre el menú contextual selecciona la opción de añadir etiqueta al álbum.
- 3.El sistema presenta la lista de etiquetas disponibles.
- 4.El usuario arrastra la etiqueta seleccionada sobre el álbum seleccionado.
- 5.El álbum dispone en este momento de una nueva etiqueta.
- 6.Si el árbol de álbumes no tenía la etiqueta se añade al final del mismo.

Condiciones

El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber selecciona un álbum de los disponibles en el sistema	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

8.Borrar Etiqueta Álbum

El usuario dispone de la posibilidad de poder borrar etiquetas de álbumes

Flujo de Eventos

Flujo básico

- 1.El usuario selecciona el álbum sobre el que quiere eliminar la etiqueta.
- 2.El usuario sobre el menú contextual selecciona la opción de borrar etiqueta del álbum.
- 3.El álbum ya no dispone en este momento de la etiqueta.
- 4.Se refresca la zona de álbumes para que el álbum no aparezca asociado a la etiqueta.

Condiciones

El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber selecciona un álbum de los disponibles en el sistema	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

9. Ver Imágenes Álbum

El usuario dispone de la posibilidad de poder ver las imágenes disponibles en un álbum.

<u>Flujo de Eventos</u>	
<i>Flujo básico</i>	
1.El usuario selecciona el álbum en la lista de álbumes. 2.El sistema accede a la Base de Datos y recupera la lista de imágenes. 3.El sistema presenta la lista de imágenes para que el usuario pueda visualizarlas.	
<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>

10. Añadir Imágenes Álbum

El usuario dispone de la posibilidad de poder añadir imágenes a un álbum

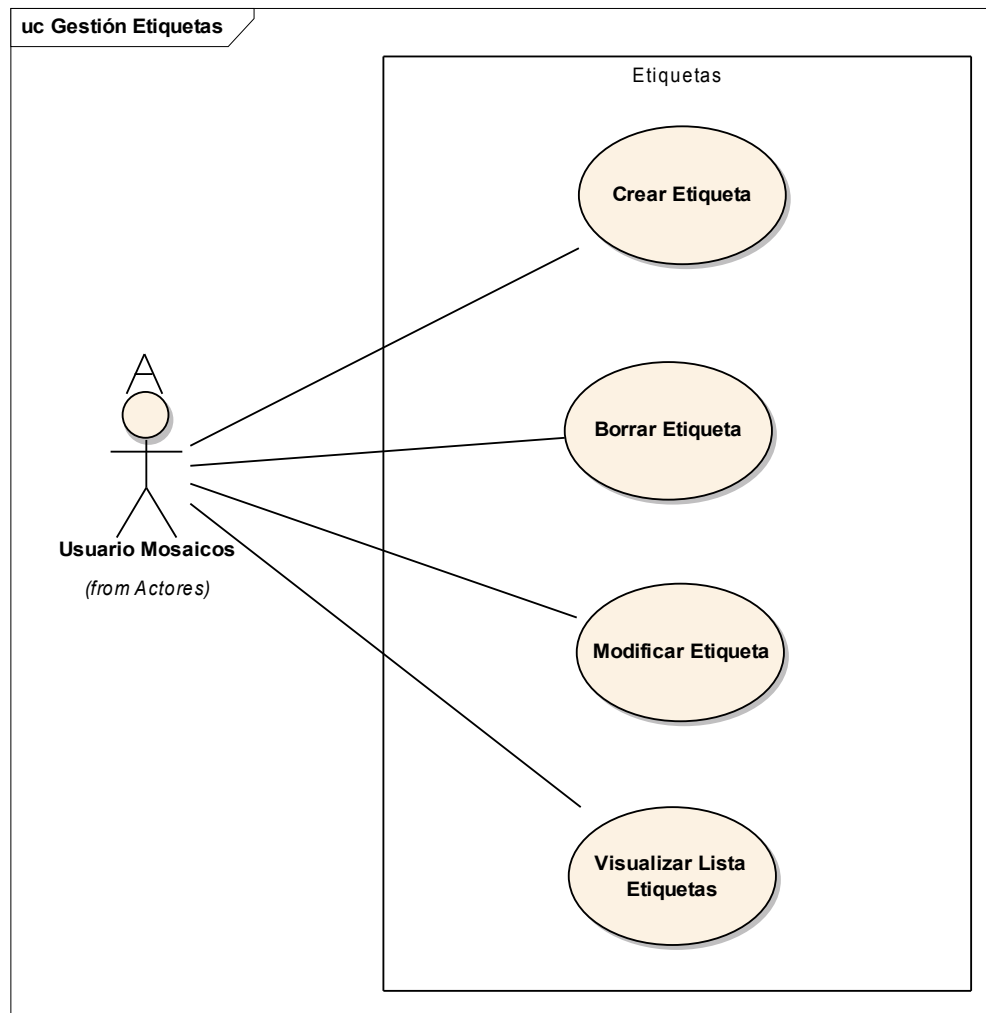
<u>Flujo de Eventos</u>	
<i>Flujo básico</i>	
1.El usuario selecciona el álbum sobre el que quiere añadir imágenes. 2.El usuario sobre el menú contextual selecciona la opción de añadir imágenes al álbum. 3.El sistema presenta un diálogo para que el usuario pueda seleccionar imágenes desde su ordenador 4.El usuario pulsa el botón de upload imágenes y las imágenes se suben al servidor.	
<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber selecciona un álbum de los disponibles en el sistema	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

11. Borrar Imágenes Álbum

El usuario dispone de la posibilidad de poder borrar las imágenes disponibles en un álbum.

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber selecciona un álbum de los disponibles en el sistema	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

4.2.3.3 Gestión Etiquetas



12. Crear Etiqueta

El usuario dispone de la posibilidad de generar etiquetas para posteriormente asignárselas a los álbumes.

Flujo de Eventos

Flujo básico

- 1.El usuario se sitúa sobre el nodo catálogo de etiquetas.
- 2.Una vez seleccionado el nodo sobre el menú contextual que le aparece al usuario selecciona la opción de creación de etiquetas.
- 3.El sistema automáticamente coloca al final del árbol la etiqueta generada. El nombre de la etiqueta que se genera tiene un nombre aleatorio que posteriormente el usuario puede cambiar
- 4.El usuario visualmente puede ver la etiqueta creada

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

13.Borrar Etiqueta

El usuario dispone de la posibilidad de poder borrar etiquetas

<u>Flujo de Eventos</u>
<i>Flujo básico</i>
<ol style="list-style-type: none"> 1.El usuario se sitúa sobre el nodo a borrar. 2.Una vez seleccionado el nodo sobre el menú contextual que le aparece al usuario selecciona la opción de borrar la etiqueta seleccionada. 3.El sistema borra la etiqueta y refresca la lista de etiquetas del usuario 4.El usuario visualmente puede ver la etiqueta borrada.

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

14.Modificar Etiqueta

El usuario tendrá la posibilidad de poder modificar el nombre de una etiqueta

Flujo de Eventos

Flujo alternativo

1. Si el nombre de la etiqueta ya existe el sistema vuelve a colocar el cursor sobre la etiqueta hasta que el usuario decida seleccionar un nuevo nombre.
2. Si desea cancelar pulsando el botón de escape el nombre de la etiqueta se deja con el nombre original.

Flujo básico

1. El usuario se sitúa sobre el nodo a modificar.
2. Una vez seleccionado el nodo sobre el menú contextual que le aparece al usuario este selecciona la opción de modificar la etiqueta seleccionada.
3. El sistema presenta al usuario un cuadro de texto sobre el nodo a modificar para que el usuario puede introducir el nuevo nombre de la etiqueta.
4. El sistema modifica el nombre de la etiqueta en la Base de Datos.
5. El usuario visualmente puede ver el nuevo nombre de la etiqueta

15. Visualizar Lista Etiquetas

El usuario tendrá la posibilidad de visualizar la lista de etiquetas disponibles en el sistema

Flujo de Eventos

Flujo básico

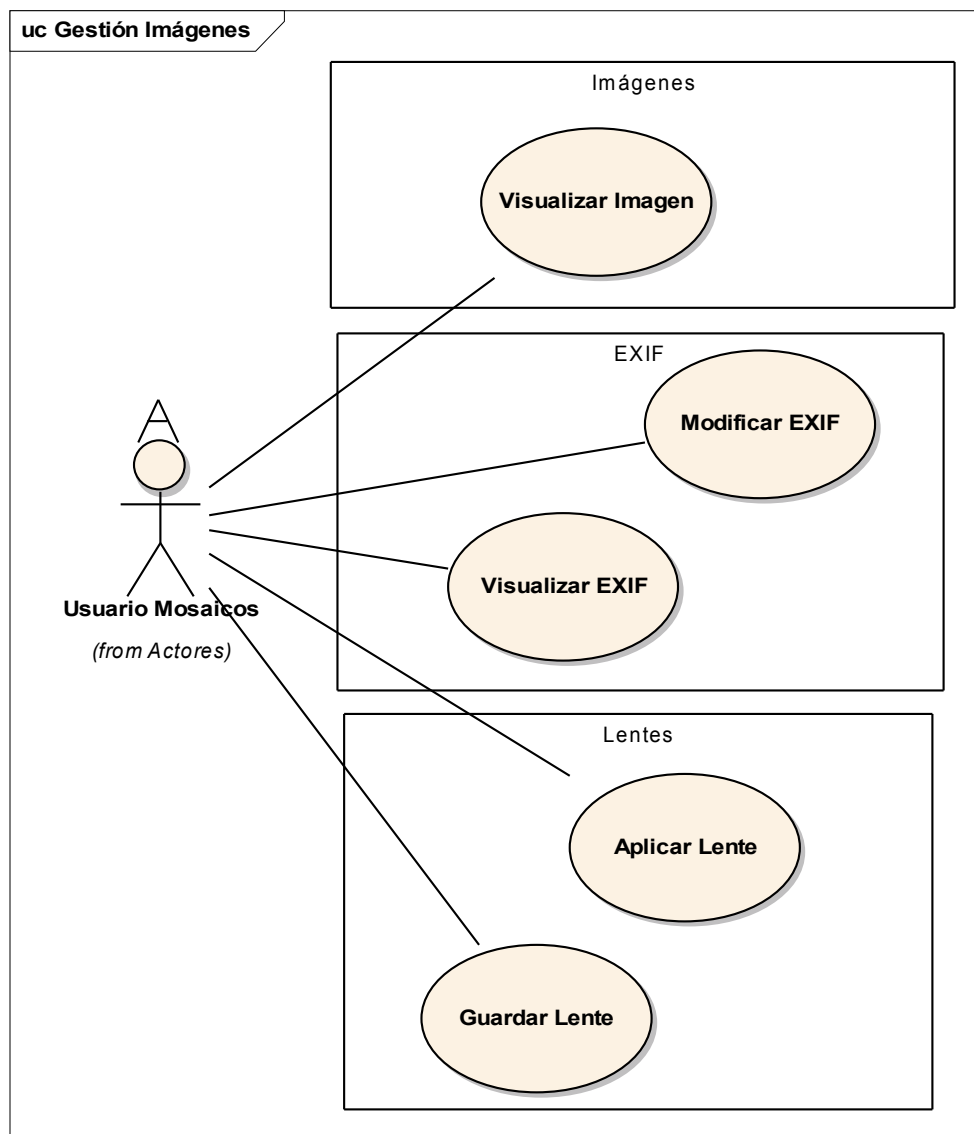
1. El usuario dispone de un enlace para refrescar la lista de etiquetas disponibles en el sistema
2. Una vez seleccionado ese enlace el sistema accede a la Base de Datos para obtener la lista de etiquetas del usuario.

Condiciones

El usuario debe estar autenticado en el sistema

Pre-Condición

4.2.3.4 Gestión Imágenes



16. Visualizar Imagen

El usuario dispone de la posibilidad de poder visualizar la imagen seleccionada.

<u>Flujo de Eventos</u>	
<i>Flujo básico</i>	
1.El usuario se sitúa sobre la imagen seleccionada 2.El sistema obtiene la información de la imagen (metadatos) y la presenta en pantalla.	

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber selecciona una imagen de los disponibles en el sistema	<u>Pre-Condición</u>

17.Modificar EXIF

El usuario dispone de la posibilidad de poder modificar los metadatos de la imagen.

<u>Flujo de Eventos</u>	
<i>Flujo básico</i>	
1.El usuario selecciona la imagen de la que quiere modificar la información de metadatos. 2.El usuario modifica unos de los valores disponibles como metadatos de la imagen 3.El sistema aplica los cambios sobre la imagen.	

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber selecciona una imagen de los disponibles en el sistema	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

18.Visualizar EXIF

El usuario dispone de la posibilidad de visualizar los metadatos de una imagen.

<u>Flujo de Eventos</u>	
<i>Flujo básico</i>	

- 1.El usuario selecciona la imagen de la que quiere visualizar la información de metadatos.
- 2.El usuario visualiza la información en una serie de formularios personalizados.

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber selecciona una imagen de los disponibles en el sistema	<u>Pre-Condición</u>

19. *Aplicar Lente*

El usuario dispone de la posibilidad de poder aplicar unos metadatos almacenados (lente) a una imagen seleccionada

<u>Flujo de Eventos</u>
<i>Flujo básico</i>
<ol style="list-style-type: none"> 1.El usuario selecciona una imagen de la que quiere modificar la información de metadatos. 2.El usuario selecciona una lente que viene con una serie de metadatos predefinidos. 3.El usuario aplica los cambios. 4.La imagen dispone de una serie de metadatos específicos.

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber selecciona una imagen de los disponibles en el sistema	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

20. *Guardar Lente*

El usuario dispone de la posibilidad de poder almacenar los metadatos en una lente.

<u>Flujo de Eventos</u>
<i>Flujo básico</i>
1.El usuario selecciona una imagen y sus metadatos se visualizan en un formulario personalizado

-
- 2.El usuario pulsa el botón de guardar lente.
 - 3.Se crea una nueva lente que permite aplicar los metadatos a otras imágenes.

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario debe haber selecciona una imagen de los disponibles en el sistema	<u>Pre-Condición</u>
La modificación se almacenará en la Base de Datos	<u>Post-Condición</u>

4.2.3.5 Gestión Mosaicos

En la Figura 36 se pueden observar los casos de uso de la gestión de mosaicos

21.Crear Mosaico.

El usuario dispone de la posibilidad de poder generar un mosaico con un conjunto de imágenes

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario deberá haber seleccionado un conjunto de imágenes	<u>Pre-Condición</u>
El usuario deberá haber subido las imágenes al servidor para su procesamiento	<u>Pre-Condición</u>
El usuario deberá haber realizado el proceso previo de alineación	<u>Pre-Condición</u>
La imagen se generará en disco para su posterior visualización	<u>Post-Condición</u>

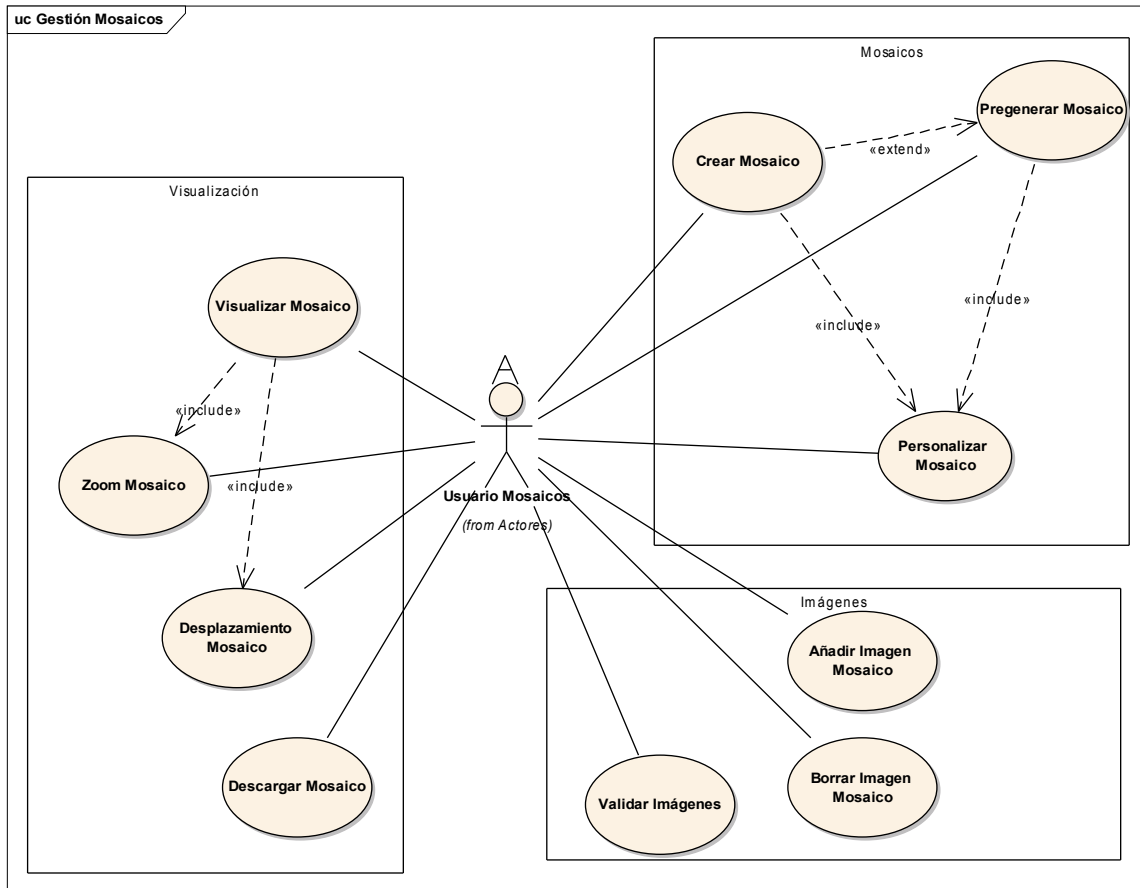


Figura 36: Gestión de mosaicos

22. Visualizar Mosaico

El usuario tendrá la posibilidad de poder visualizar la imagen generada.

Condiciones	
El usuario debe estar autenticado en el sistema	Pre-Condición
El usuario debe haber generado previamente un mosaico	Pre-Condición

23. Personalizar Mosaico

El usuario tendrá la posibilidad de poder personalizar el proceso de generación de mosaico mediante la modificación de los metadatos de las imágenes.

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario deberá haber seleccionado un conjunto de imágenes	<u>Pre-Condición</u>

24. *Pregenerar Mosaico*

El usuario dispone de la posibilidad de poder pre-generar un mosaico con un conjunto de imágenes previamente seleccionadas.

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario deberá haber seleccionado un conjunto de imágenes	<u>Pre-Condición</u>
El usuario deberá haber subido las imágenes al servidor para su procesamiento	<u>Pre-Condición</u>
La imagen se generará en disco para su posterior visualización	<u>Pre-Condición</u>

25. *Añadir Imagen Mosaico*

El usuario dispondrá de la posibilidad de poder añadir una imagen al mosaico

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>

26. *Borrar Imagen Mosaico*

El usuario dispondrá de la posibilidad de poder borrar una imagen del mosaico de imágenes

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario deberá haber seleccionado un conjunto de imágenes	<u>Pre-Condición</u>

27. Validar Imágenes

El usuario dispondrá de la posibilidad de poder validar las imágenes seleccionadas por el usuario con el objetivo de verificar su idoneidad para la composición del mosaico

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario deberá haber seleccionado un conjunto de imágenes	<u>Pre-Condición</u>

28. Zoom Mosaico

El usuario podrá realizar un zoom (más y menos) sobre la imagen generada

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario deberá haber generado el mosaico	<u>Pre-Condición</u>

29. Desplazamiento Mosaico

El usuario deberá poder desplazarse dentro de la imagen generada

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario deberá haber generado el mosaico	<u>Pre-Condición</u>

30. Descargar Mosaico

El usuario tendrá la posibilidad de poder descargar la imagen generada al PC del usuario.

<u>Condiciones</u>	
El usuario debe estar autenticado en el sistema	<u>Pre-Condición</u>
El usuario deberá haber generado el mosaico	<u>Pre-Condición</u>

4.3 Matriz de Trazabilidad

La matriz de trazabilidad permite verificar que los requisitos previamente identificados están cubiertos por todo los casos de uso. Es una herramienta muy útil para tener una trazabilidad completa de los desarrollos realizados.

En las siguientes ilustraciones se puede ver como se combinan los requisitos de usuarios con los casos de uso a implementar. En las tablas que se van a presentar las filas de la izquierda hacen referencia a los distintos requisitos de usuarios mientras que las columnas hacen referencia a los casos de uso. En el punto de intersección entre un requisito y un caso de uso aparece una flecha que indica que el requisito tiene un caso de uso relacionado.

4.3.1 Gestión de usuarios

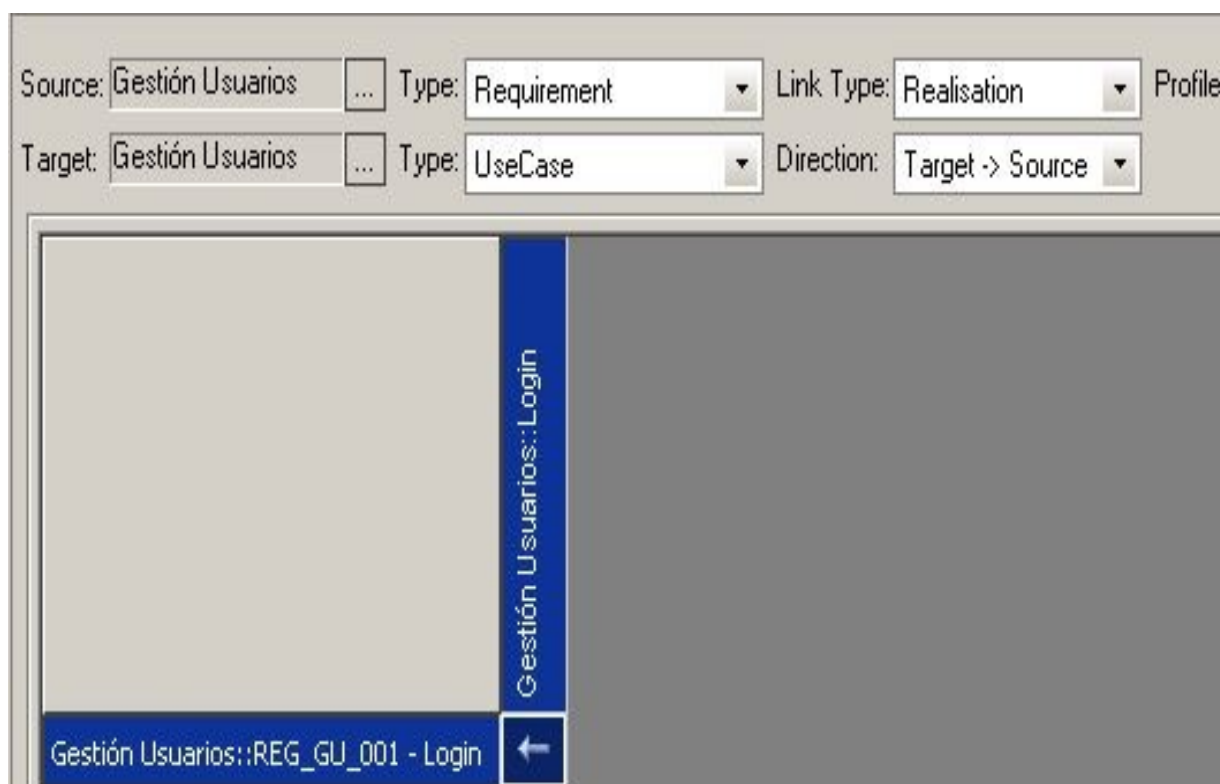


Figura 37: Matriz de Trazabilidad. Gestión de usuarios

4.3.2 Gestión de álbumes

Source: Gestión Álbums ...	Type: Requirement	Link Type: Realisation	Profile:
Target: Gestión Álbums ...	Type: UseCase	Direction: Target -> Source	
	Gestión Álbums: Añadir Etiqueta	Gestión Álbums: Añadir Imagen	Gestión Álbums: Borrar Álbum
Gestión Álbums::REG_GA_000 - Gestionar Álbums	←	←	←
Gestión Álbums::REG_GA_002 - Borrar Álbum		←	
Gestión Álbums::REG_GA_006 - Borrar Etiqueta Álbum			←
Gestión Álbums::REG_GA_007 - Ver Imagenes Álbum			←
Gestión Álbums::REG_GA_008 - Añadir Imagenes Álbum		←	
Gestión Álbums::REG_GA_009 - Borrar Imagenes Álbum			←
Gestión Álbums::REQ_GA_001 - Crear Álbum			←
Gestión Álbums::REQ_GA_003 - Modificar Álbum			←
Gestión Álbums::REQ_GA_004 - Visualizar Lista Álbums			←
Gestión Álbums::REQ_GA_005 - Añadir Etiqueta Álbum	←		

Figura 38: Matriz de Trazabilidad. Gestión de Álbumes

4.3.3 Gestión de etiquetas

Source: Gestión Etiquetas ...	Type: Requirement	Link Type: Realisation	Profile:
Target: Gestión Etiquetas ...	Type: UseCase	Direction: Target -> Source	
	Gestión Etiquetas: Borrar Etiqueta	Gestión Etiquetas: Crear Etiqueta	Gestión Etiquetas: Modificar Etiqueta
Gestión Etiquetas::REG_GE_000 - Gestionar Etiquetas	←	←	←
Gestión Etiquetas::REG_GE_002 - Borrar Etiqueta	←		
Gestión Etiquetas::REQ_GE_001 - Crear Etiqueta		←	
Gestión Etiquetas::REQ_GE_003 - Modificar Etiqueta			←
Gestión Etiquetas::REQ_GE_004 - Visualizar Etiquetas			←

Figura 39: Matriz de Trazabilidad. Gestión de Etiquetas

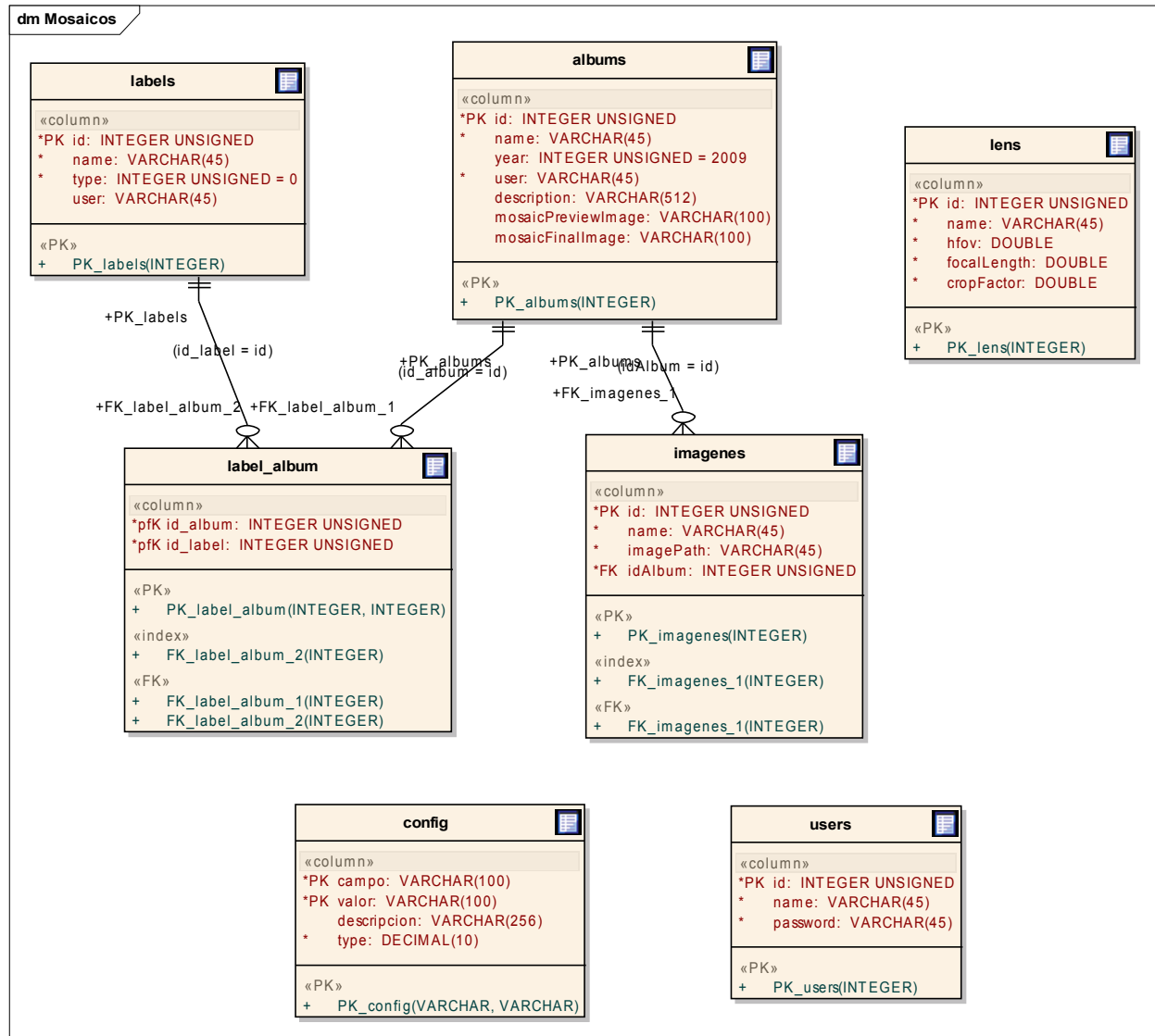
4.3.4 Gestión de imágenes

Source:	Gestión Imágenes	...	Type:	Requirement		Link Type:	Realisation		Pro
Target:	Gestión Imágenes	...	Type:	UseCase		Direction:	Target -> Source		

Figura 40: Matriz de Trazabilidad. Gestión de Imágenes

4.4 Diseño de la base de datos

Este apartado recoge las tablas de la base de datos del sistema mostrando todos sus atributos, los tipos de éstos y las relaciones que hay entre dichas tablas.



Las tablas disponibles en el sistema son las siguientes:

- **Albums:** Almacenan los álbumes de imágenes disponibles en el sistema
- **Config:** Información de configuración de la aplicación.
- **Imágenes:** Almacena la información de las imágenes disponibles en cada álbum
- **Label_álbum:** Almacena la relación entre las etiquetas y los álbumes disponibles en el sistema.
- **Labels:** Almacena la lista de etiquetas disponibles en el sistema.
- **Lens:** Almacena la información de las lentes disponibles en el sistema.

- **Users** : Almacena la lista de usuarios del sistema.

4.4.1 álbumes

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	id	INTEGER	True	True	10	0	0		Identificador del álbum
False	name	VARCHAR	True	False	45	0	0		Nombre del álbum
False	year	INTEGER	False	False	10	0	0	2009	Año del álbum
False	user	VARCHAR	True	False	45	0	0		Usuario propietario del álbum
False	description	VARCHAR	False	False	512	0	0		Descripción del álbum
False	mosaicPreviewImage	VARCHAR	False	False	100	0	0		Nombre de la imagen de previsualización
False	mosaicFinalImage	VARCHAR	False	False	100	0	0		Nombre de la imagen final

Constraints

Name	Type	Columns	Initial Code	Notes
PK_albums	Public	id		

Relationships

Columns	Association	Notes
(id_álbum = id)	0..* label_album.FK_label_álbum_1 1 albums.PK_albums	
(idAlbum = id)	0..* imagenes.FK_imagenes_1 1 albums.PK_albums	

4.4.2 config

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	campo	VARCHAR	True	True	100	0	0		Identificador del campo
True	valor	VARCHAR	True	True	100	0	0		Valor del campo
False	descripcion	VARCHAR	False	False	256	0	0		Descripción del campo de configuración
False	type	DECIMAL	True	False	0	10	0		Tipo de campo

Constraints

Name	Type	Columns	Initial Code	Notes
PK_config	Public	campo valor		

4.4.3 imagenes

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	id	INTEGER	True	True	10	0	0		Identificador de la imagen
False	name	VARCHAR	True	False	45	0	0		Nombre de la imagen
False	imagePath	VARCHAR	True	False	45	0	0		Path de la imagen
False	idAlbum	INTEGER	True	False	10	0	0		Identificador del álbum al que pertenece la imagen

Constraints

Name	Type	Columns	Initial Code	Notes
PK_imagenes	Public	id		
FK_imagenes_1	Public	idAlbum		
FK_imagenes_1	Public	idAlbum		

Relationships

Columns	Association	Notes
(idAlbum = id)	0..* imagenes.FK_imagenes_1 1 albums.PK_albums	

4.4.4 label_album

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	id_album	INTEGER	True	True	10	0	0		Identificador del álbum
True	id_label	INTEGER	True	True	10	0	0		Identificador de la etiqueta

Constraints

Name	Type	Columns	Initial Code	Notes
PK_label_album	Public	id_album id_label		
FK_label_album_2	Public	id_label		
FK_label_album_1	Public	id_album		
FK_label_album_2	Public	id_label		

Relationships

Columns	Association	Notes
(id_label = id)	0..* label_album.FK_label_album_2 1 labels.PK_labels	
(id_album = id)	0..* label_album.FK_label_album_1 1 albums.PK_albums	

4.4.5 labels

Database: MySQL, *Stereotype:* «table», *Paquete:* Mosaicos
Detail: Created on 17/10/2009. Last modified on 17/10/2009.
Notes:

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	id	INTEGER	True	True	10	0	0		Identificador de etiqueta
False	name	VARCHAR	True	False	45	0	0		Nombre de la etiqueta
False	type	INTEGER	True	False	10	0	0	0	Tipo de la etiqueta 0. Publica, 1. Privada
False	user	VARCHAR	False	False	45	0	0		Usuario propietario de la etiqueta

Constraints

Name	Type	Columns	Initial Code	Notes
PK_labels	Public	id		

Relationships

Columns	Association	Notes
(id_label = id)	<u>0..*</u> label_album.FK_label_album_2 <u>1</u> labels.PK_labels	

4.4.6 lens

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	id	INTEGER	True	True	10	0	0		Identificador de la lente
False	name	VARCHAR	True	False	45	0	0		Nombre de la lente
False	hfov	DOUBLE	True	False	0	0	0		Valor de hfov
False	focalLength	DOUBLE	True	False	0	0	0		Valor de distancia focal
False	cropFactor	DOUBLE	True	False	0	0	0		Valor de factor de recorte

Constraints

Name	Type	Columns	Initial Code	Notes
PK_lens	Public	id		

4.4.7 users

Database: MySQL, *Stereotype:* «table», *Paquete:* Mosaicos
Detail: Created on 17/10/2009. Last modified on 17/10/2009.
Notes:

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
True	id	INTEGER	True	True	10	0	0		Identificador del usuario
False	name	VARCHAR	True	False	45	0	0		Nombre del usuario

False	password	VARCHAR	True	False	45	0	0		Password del usuario
-------	----------	---------	------	-------	----	---	---	--	----------------------

Constraints

<u>Name</u>	<u>Type</u>	<u>Columns</u>	<u>Initial Code</u>	<u>Notes</u>
PK_users	Public	id		

5 Tecnologías utilizadas

Antes de desarrollar el diseño, hay que tener en cuenta todas las implicaciones que se derivan del entorno y sus restricciones.

La primera restricción y quizá la más importante es que tiene que estar disponible en el entorno de Internet para ser accesible por cualquier usuario con un navegador web. Esto implica, que ha de ser diseñado para adecuarse al entorno de Internet.

La elección de la herramienta de generación de mosaicos que se ha comentado en los apartados anteriores es también un punto importante a la hora de elegir la tecnología con la que se va a desarrollar la aplicación.

El tipo de aplicación que se ha desarrollado dispone de dos partes completamente diferenciadas. Por una parte se dispone de un interfaz de usuario web muy potente y compatible con la extensa mayoría de navegadores del mercado y por otra de un módulo servidor que permite realizar todas las operaciones más pesadas y que necesitan de una mayor capacidad computacional.

En los siguientes apartados se profundizará más en detalle en las distintas tecnologías utilizadas indicando el por qué se han elegido para el desarrollo del proyecto indicando sus ventajas e inconvenientes.

Se comentará también las tecnologías de mantenimiento de versiones así como de despliegue de la aplicación.

5.1 Tecnologías de programación

La arquitectura del proyecto se puede ver en la .Figura 41.

Partiendo de esta arquitectura base y observando el tipo de módulos a desarrollar se puede analizar cuales son las distintas posibilidades que se han podido tomar como punto de partida para el desarrollo de la aplicación.

5.1.1 Cliente

Como se ha comentado en el apartado anterior, el objetivo del proyecto es la realización de una aplicación web que pueda ser ejecutada en la amplia mayoría de navegadores del mercado. Esta característica va a permitir que la aplicación no necesite de unos recursos específicos en el PC del cliente permitiendo que un gran porcentaje de los usuarios sino la mayoría vayan a poder utilizar los servicios proporcionados por dicho sistema.

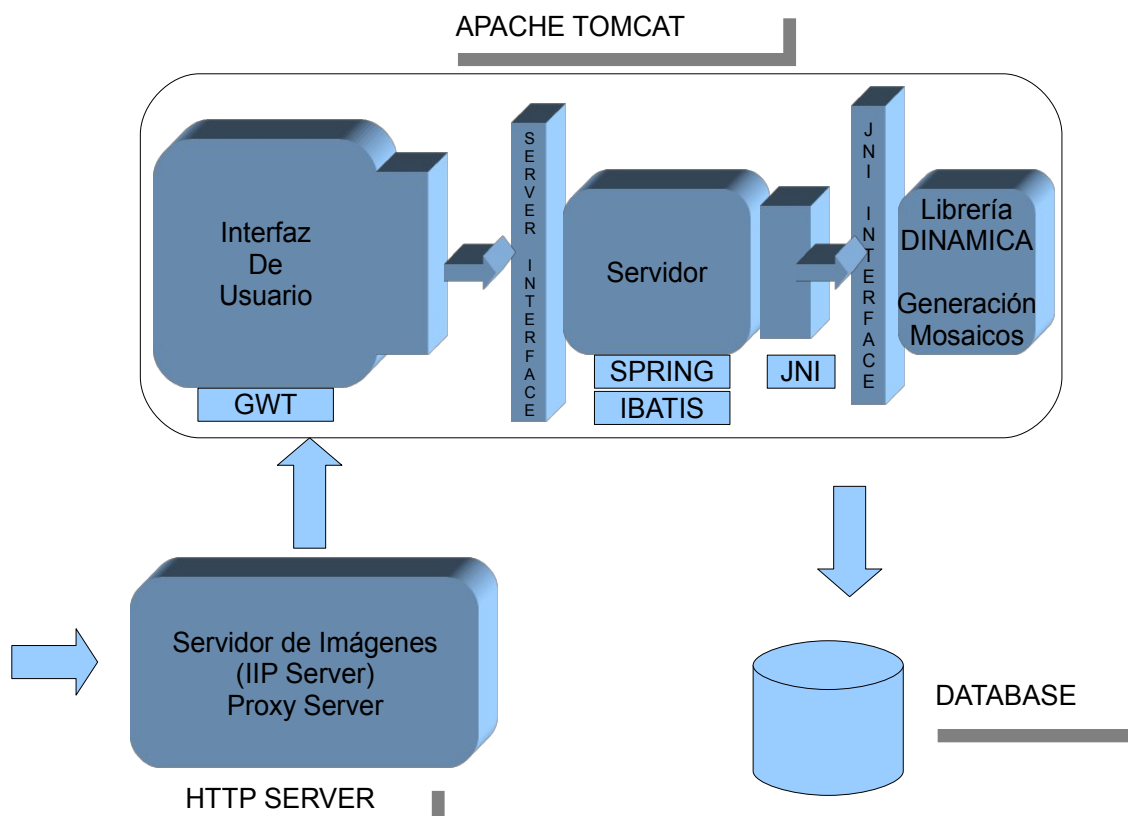


Figura 41: Arquitectura del proyecto

También posibilita que los usuarios no tengan que descargarse ningún tipo de software en su ordenador, necesidad que en algunos casos provoca reticencias en los usuarios por posibles problemas de seguridad.

Uno de los principales problemas de las aplicaciones web, al contrario que las aplicaciones de escritorio, es la compatibilidad entre los navegadores. Actualmente en el mercado existen un número creciente de navegadores entre los que podemos destacar los siguientes:

- Internet Explorer.
- Firefox
- Safari
- Opera
- Netscape
- Chrome

Durante la evolución de la Web ha quedado patente la necesidad de disponer de estándares para sentar las bases del desarrollo de aplicaciones Web y el fomento de la red. Entre estos estándares podemos destacar el HTML, Javascript, hojas de estilo (CSS), AJAX, etc.

Aunque la estandarización es un punto que ha estado siempre en mente en la implementación de

estos estándares en los distintos navegadores, la competitividad ha provocado que los distintos fabricantes dispongan de extensiones que permiten mejorar o optimizar la visualización de las páginas Web. Esto provoca que determinados sitios web se vean mejor con un navegador que con otro o que incluso sean inoperativos con alguno de ellos.

Durante los últimos años con el auge de la Web 2.0 las capacidades de los navegadores han aumentado proporcionalmente al número de usuarios que ha empezado a utilizar sus características. Este aumento de aplicaciones ha provocado un incremento del número de librerías utilizadas para desarrollar componentes para estas aplicaciones.

En los últimos años ha surgido una nueva tecnología cuyo objetivo es solucionar todos estos problemas facilitando el desarrollo de aplicaciones Web. Esta tecnología pretende que los desarrolladores de aplicaciones se concentren principalmente en las funcionalidades que desean que dispongan sus aplicaciones permitiendo trabajar con componentes gráficos similares a swing/awt.

Google Web Toolkit (GWT) es un framework de desarrollo en **Java** de código abierto, que evita tener que conocer toda la cantidad de productos y tecnologías disponibles para escribir aplicaciones AJAX, las cuales son difíciles de manejar y propensas a errores. La idea sobre la que se sustenta GWT es que el desarrollo de aplicaciones se realiza utilizando el lenguaje de programación Java y el entorno de desarrollo IDE preferido.

GWT, una vez realizada la programación, dispone de herramientas para compilar y traducir el código Java a un lenguaje HTML en el que se incluye Javascript y hojas de estilo. GWT proporciona las adaptaciones para cada navegador permitiendo que el código HTML sea ejecutado en cualquier navegador del mercado.

Para más información ver [GWT].

5.1.2 Servidor

El módulo de servidor desarrollado para este proyecto permite que la aplicación cliente pueda disponer de una serie de servicios destinados por una parte al manejo de los contenidos gestionados por la aplicación y por otra parte a la generación de mosaicos con las imágenes cargadas en este sistema.

El módulo de servidor se ha desarrollado al igual que el cliente utilizando tecnología Java de servidor. Este módulo se distribuye como un fichero WAR (War File) el cual se despliega sobre un servidor de aplicaciones en este caso sobre el contenedor de servlets tomcat.

Para la realización de este módulo servidor se han utilizado entre otras, tres tecnologías que pretenden estandarizar el proceso de desarrollo de aplicaciones Web permitiendo que los desarrolladores concentren toda su atención en las funcionalidades de la aplicación y dejen a estos framework el resto de operativa.

- JNI.
- Spring.
- Ibatis.

5.1.2.1 JNI

JNI es un mecanismo que nos permite ejecutar código nativo desde Java y viceversa. El código nativo son funciones escritas en un lenguaje de programación como C o C++ para un sistema operativo donde se está ejecutando la máquina virtual

Se usa principalmente para poder escribir métodos nativos que permitan solventar situaciones en las que una aplicación no puede ser enteramente escrita en Java, o cuando se necesite utilizar funciones ofrecidas por librerías escritas en otros lenguajes de programación.

5.1.2.2 Spring

Spring es un framework de código abierto para el desarrollo de aplicaciones para la plataforma Java. Proporciona una serie de características entre las que podemos destacar las siguientes:

- Gestión de configuración basada en JavaBeans, aplicando los principios de Inversión de Control (IoC). Esto hace que la configuración de aplicaciones sea rápida y sencilla.
- Ofrece una capa genérica de abstracción para la gestión de transacciones, permitiendo gestores de transacción fácilmente acoplables, y haciendo sencilla la definición de transacciones sin tratarlas a bajo nivel
- Ofrece una capa de abstracción JDBC que ofrece una significativa jerarquía de excepciones (evitando la necesidad de obtener de SQLException los códigos que cada gestor de base de datos asigna a los errores), simplifica el manejo de errores, y reduce considerablemente la cantidad de código necesario.
- Integración con Hibernate, JDO e iBatis SQL Maps en términos de soporte a implementaciones DAO y estrategias con transacciones.
- Funcionalidad AOP, totalmente integrada en la gestión de configuración de Spring. Se puede aplicar AOP a cualquier objeto gestionado por Spring, añadiendo aspectos como gestión de transacciones declarativa.
- Un framework MVC (*Model-View-Controller*), construido sobre el núcleo de Spring. Este framework es altamente configurable vía interfaces y permite el uso de múltiples tecnologías para la capa vista.

5.1.2.3 iBatis

iBatis es un framework de persistencia para Java. Es un framework ligero que facilita el diseño de la capa de persistencia utilizada en las aplicaciones Java para acceder a nuestro repositorio de datos.

Permite que el desarrollador se olvide de la implementación del acceso a datos, únicamente se debe preocupar por realizar una correcta configuración.

Ibatis no es un ORM (Mapeo Objeto Relacional) al contrario que por ejemplo Hibernate, por tanto en iBatis el SQL lo tendremos que escribir nosotros.

Sus características más importantes son.

- Ejecuta el SQL escrito por nosotros mediante JDBC (Java DataBase Connectivity), por lo que nos olvidamos de los múltiples try/catch y de los cierres específicos de las conexiones.
- Mapea propiedades de objetos a parámetros para las PreparedStatement..
- Mapea los resultados de una query a un objeto o una lista de objetos permitiendo posteriormente trabajar directamente con objetos Java.
- Fuerte integración con Spring.

iBatis proporciona básicamente un modo simple y flexible de mover los datos entre los objetos Java y la base de datos relacional. En resumen, mapeos relacionales a objetos.

De este modo, se tiene toda la potencia de SQL sin necesidad de escribir una línea de código JDBC. SQL Maps reduce considerablemente la cantidad de código necesario para acceder a una base de datos relacional. Este framework mapea la clases a sentencias SQL usando un descriptor XML muy simple. Los pasos que se realizan a la hora de utilizar *SQL Maps* son lo siguientes:

- Crear los *ficheros de mapeos* correspondientes. Los ficheros de mapeos son ficheros XML que contiene nuestro código SQL para realizar determinadas operaciones con sus parámetros de entrada y salida correspondientes.
- Crear los *DTOs* equivalentes. Van a contener los parámetros de entrada y de salida que hemos definido previamente en los ficheros de mapeos. Normalmente estos objetos se corresponderán con la/s tabla/s correspondientes de nuestro Modelo de Datos.
- Crear el *código Java* trabajando con los ficheros de mapeos.

5.2 Tecnologías de desarrollo

5.2.1 Eclipse IDE SDK

Eclipse es un Entorno Integrado de Desarrollo (IDE), para todo tipo de aplicaciones libres, inicialmente desarrollado por IBM, y actualmente gestionado por la Fundación Eclipse.

Es una herramienta para el programador desarrollada principalmente para el desarrollo de aplicaciones Java, facilitando al máximo la gestión de proyectos colaborativos mediante el control de versiones CVS o SVN.

Es posible añadir nuevas funcionalidades al editor, a través de nuevos módulos ('plugins'), para programar en otros lenguajes de programación además de Java como C/C++, PHP, Python, Ruby, Cobol, ...y más.

Entre sus características más importante están las siguientes:

- Multiplataforma (GNU/Linux, Solaris, Mac OSX, Windows)
- Soportado para distintas arquitecturas (x86, 64, ...)
- Estructura de plug-in que hace sencillo añadir nuevas características y funcionalidades.
- Control de versiones con cvs o con subversión (con subclipse). Resaltado de sintaxis, autocompletado, tabulador de un bloque de código seleccionado, ... es decir, gran cantidad de

utilidades de edición que ayudan enormemente al programador.

- Asistentes (wizards) para la creación, exportación e importación de proyectos; para generar esqueletos de códigos (templates), etc.

5.3 Tecnologías de despliegue

5.3.1 Maven

Maven es una herramienta de gestión de información de proyectos. Maven está basado en el concepto de un modelo de objetos del proyecto POM (*Project Object Model*) en el que todos los productos (*artifacts*) generados por Maven son el resultado de consultar un modelo de proyecto bien definido. Compilaciones, documentación, métricas sobre el código fuente y un innumerable número de informes son todos controlados por el POM.

Maven tiene muchos objetivos, pero resumiendo Maven intenta hacer la vida del desarrollador sencilla proporcionando una estructura de proyecto bien definida, unos procesos de desarrollo bien definidos a seguir, y una documentación coherente que mantiene a los desarrolladores y clientes informados sobre lo que ocurre en el proyecto. Maven aligera en gran cantidad lo que la mayoría de desarrolladores consideran trabajo pesado y aburrido y les permite proseguir con la tarea. Esto es esencial en proyectos *open source* donde no hay mucha gente dedicada a la tarea de documentar y propagar la información crítica sobre el proyecto que es necesaria para atraer potenciales nuevos desarrolladores y clientes.

La ambición de Maven es hacer que el desarrollo interno del proyecto sea altamente manejable con la esperanza de proporcionar más tiempo para el desarrollo de los mimos.

Características:

- El modelo de objetos del proyecto POM es la base de cómo Maven trabaja. El desarrollo y gestión del modelo está controlado desde el modelo del proyecto
- Un único conjunto de métodos son utilizados para todos los proyectos que se gestionan. Ya no hay necesidad de estar al tanto de innumerables sistemas de compilación. Cuando las mejoras se hacen en Maven todos los usuarios se benefician.
- Publicación del sitio web basado en el POM. Una vez el POM es exacto los desarrolladores pueden publicar fácilmente el contenido del proyecto, incluyendo la documentación personalizada más el amplio conjunto de documentación generada por Maven a partir del código fuente.
- Publicación de distribuciones basada en el POM.
- Maven se basa en el uso de un repositorio central de librerías, utilizando un mecanismo que permite descargar automáticamente aquellas necesarias en el proyecto, lo que permite a los usuarios de Maven reutilizar librerías entre proyectos y facilita la comunicación entre proyectos para asegurar que la compatibilidad entre distintas versiones es correctamente tratada.
- Guías para la correcta disposición de los directorios. Maven contiene documentación sobre

como disponer los directorios de forma que una vez es aprendida se puede ver fácilmente cualquier otro proyecto que use Maven.

5.4 Tecnologías de ejecución

El software requerido para la ejecución del sistema es el siguiente:

1. Servidor de aplicaciones web (Tomcat)
2. Servidor Apache para el soporte cgi
3. Sistema Gestor de base de datos relacional (MySQL)
4. Navegador web (Internet Explorer, Mozilla, Netscape Navigator, Opera, ...)

5.4.1 Tomcat

El servidor Jakarta Apache Tomcat es el servidor Web más utilizado a la hora de trabajar con Java en entornos web. Este servidor funciona como un contenedor de servlets y JSP.

Los servlets son objetos que se ejecutan dentro de un contexto de un contenedor y que permiten extender la funcionalidad del mismo proporcionando una serie de servicios a los usuarios. El uso más común de los servlets es servir de punto de entrada a las peticiones enviadas por los usuarios y destinadas a ofrecer un resultado bien en formato de página web (HTML) o en otro tipo de formato de intercambio (XML,...).

El servidor tomcat implementa las especificaciones de servlets y de Java Server Pages (JSP) de Sun Microsystems.

Entre sus características más importantes están las siguientes.

- Facilidad de instalación.
- Arquitectura modular.
- Multiplataforma: Se ejecuta en diversas plataformas tales como Windows, Linux, etc.
- Múltiples conectores para extender sus funcionalidades.
- Puede ser usado como pequeño servidor para probar páginas JSP y servlets, o puede integrarse en el servidor Web Apache Tomcat.

5.4.2 Apache

El servidor Apache es un servidor de red para el protocolo HTTP, que funciona como un proceso standalone, y que permite servir contenido de diverso tipo hacia el usuario.

Este servidor dispone de una arquitectura modular que permite extender el servidor para incluir nuevas características a base de plugins.

Entre sus características más importantes están las siguientes.

- Multiplataforma: Se ejecuta en diversas plataformas tales como Windows, Linux, etc.
- Extensible: A través de múltiples plugins que permiten extender sus características.
- Clustering y alta disponibilidad a partir de conectores específicos.
- Virtual Host: Posibilidad de disponer de varios entornos de ejecución dentro de un mismo contexto posibilitando la gestión independiente de dominios.

5.4.3 Mysql

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Se distribuye bajo licencia GPL así como bajo una licencia privativa para cuando se quiere integrar dentro de un software comercial.

Entre sus características más importantes están las siguientes.

- Facilidad de instalación.
- Escrito en C y en C++
- Funciona en diferentes plataformas.
- Uso completo de multi-threaded mediante threads del kernel. Pueden usarse fácilmente múltiples CPUs si están disponibles.
- Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
- Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
- Soporte de gran cantidad de tipos de datos para las columnas.

6 Manual de usuario

Este capítulo incluye información sobre el funcionamiento de la aplicación Web con el objetivo de que el usuario la utilice como guía de referencia para un correcto uso del sistema.

La aplicación está destinada a usuarios finales y pretende que dichos usuarios puedan realizar una composición en formato mosaico de un conjunto de imágenes que dispongan de una serie de características de afinidad.

El manual del usuario se divide en tres apartados claramente diferenciados:

- Componentes de la aplicación: Donde se indican los elementos fundamentales de la aplicación que los usuarios deben conocer para poder realizar una composición óptima.
- Estructura de la aplicación: Donde se muestra la estructura de la aplicación y la funcionalidad accesible desde cada uno de los paneles. En este apartado se indica la localización de los componentes previamente comentados y accesibles a través de los distintos paneles.
- Proceso de generación de mosaicos: Donde se explican los pasos necesarios para la generación del mosaico indicando todas las acciones que hay que realizar para llevar a cabo una composición óptima.

6.1 Componentes

La aplicación de generación de mosaicos contempla una serie de componentes básicos que son clave a la hora de poder conocer el funcionamiento de la aplicación y poder controlar el proceso de generación de un mosaico.

En este apartado se pretende mostrar una descripción básica de cada uno de estos componentes cuyas funcionalidades posteriormente se podrán conocer cuando se muestre su ubicación dentro de la herramienta y más adelante cuando se detalle el proceso de generación de un mosaico.

- Imagen: Una imagen es un archivo en un formato gráfico típico (jpg, png, gif) que muestra un vista reducida del espacio total a representar en una panorámica completa. Típicamente cuando un usuario toma una sucesión de fotos para generar una panorámica o mosaico cada imagen almacena información específica de la zona más una información de superposición que permite que programas específicos puedan enlazar dichas imágenes. Cuanto mayor es el área de superposición es más sencilla para un programa fotográfico y a menor área se necesitan algoritmos más complejos para realizar dicha composición.
- Mosaico: Un mosaico es el producto resultado de la unión de imágenes que comparten una característica de superposición común. En esta aplicación es por tanto el resultado final esperado por un usuario cuando este realizar la generación.
- Álbum: Un álbum es un concepto lógico gestionado por esta aplicación para facilitar la agrupación de imágenes similares para un procesamiento más sencillo. Este concepto al

contrario que los dos anteriores no existe en un proceso de generación de mosaicos típico pero en esta aplicación permite que los usuarios puedan agrupar imágenes bajo una carpeta común y realizar la composición de una forma más ordenada.

- Etiqueta: De la misma forma que el concepto de álbum la etiqueta no es parte importante o clave en el proceso de generación de mosaicos. La etiqueta permite agrupar álbumes con características comunes (imágenes de la luna, paisajes urbanos, paisajes rurales, etc.). De esta forma los usuarios pueden manipular los álbumes de una forma más cómoda cuando el número de elementos (álbumes) empieza a tener un volumen considerable.
- Metadatos de imagen: Los datos o metadatos de la imagen más que un elemento a considerar como los anteriores con vida propia, son la característica del proceso de alineación que, correctamente configurados, permiten generar una imagen mosaico de alta calidad y con el mayor número de coincidencias. La óptica de la imagen (exposición, factor de recorte, etc.) es el factor que posibilita la optimización del proceso de alineación.

6.2 Estructura de la aplicación

El generador de mosaicos es un aplicativo Web y por tanto es necesario disponer de un navegador compatible (Firefox, IExplore, Opera, etc.) para su utilización.

Como se ha comentado anteriormente en el apartado 5.1.1, la utilización de la tecnología Google Web Toolkit permite realizar aplicaciones cuya funcionamiento es totalmente compatible en la amplia mayoría de los navegadores del mercado así como en sus distintas versiones sin necesidad de realizar casi ningún esfuerzo en la adaptación de la misma, esfuerzo que suele ser uno de los puntos problemáticos y que más esfuerzos necesitan a la hora de desarrollar aplicaciones accesibles a través de un navegador.

El usuario para acceder a la aplicación de generación de mosaicos deberá entrar a través de la página principal de la plataforma Ciclope Astro. Este módulo es un componente independiente de la plataforma pero comparte características comunes como son el proceso de autenticación.

6.2.1 Pantalla Autenticación

Una vez que el usuario ha accedido al aplicativo, el módulo presentará un diálogo donde el usuario deberá introducir un login y una contraseña. Si el usuario no dispone de contraseña deberá dirigirse a la página principal del proyecto Ciclope Astro con el objetivo de registrarse en el sistema. Una vez obtenida esta información el sistema le presentará con una pantalla en la que deberá consignar sus credenciales.

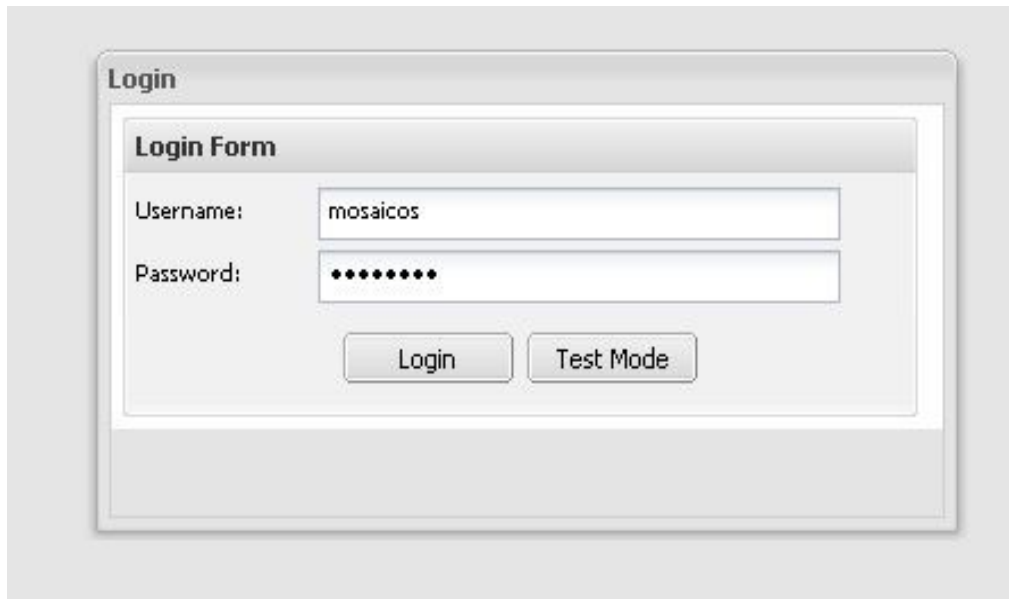


Figura 42: Pantalla Login

Una vez autenticado el usuario entra en el aplicativo a partir de cual puede realizar el proceso de alineación.

6.2.2 Pantalla principal

Una vez autenticado en el sistema el usuario accede a la pantalla principal del aplicativo, la cual está formada por cuatro pestañas, localizadas en la parte inferior a partir de las cuales se puede acceder a todas las funcionalidades de la aplicación.



Figura 43: Panel de pestañas

Estos paneles son los siguientes:

- Panel Acerca De: Donde se muestra la información básica de la aplicación.
- Panel Asistente: Donde se cargan y procesan las imágenes
- Panel de personalización de cámara: Donde se modifican los parámetros de cada una de las imágenes.
- Panel de Visualización: Donde se visualiza el resultado del proceso de generación.

En las siguientes secciones se desarrolla más en detalle el contenido de cada una de las pestañas permitiendo que el usuario conozca en profundidad cada una de ellas.

6.2.2.1 Pestaña “Acerca de”

A través de esta pantalla el usuario puede conocer el proyecto, el logo del mismo y el tipo de licencia con el que se distribuye el software. Esta pantalla es meramente descriptiva.



Figura 44: Pestaña Acerca De

6.2.2.2 Pestaña “Asistente”

Esta pestaña es la pantalla principal de la aplicación a través de la cual se realiza todo el procesamiento del módulo. Una visión general de esta pestaña se ofrece en la Figura 45.

La pantalla principal está organizada siguiendo la siguiente estructura en forma de paneles.

- Panel Superior.
- Panel Inferior.

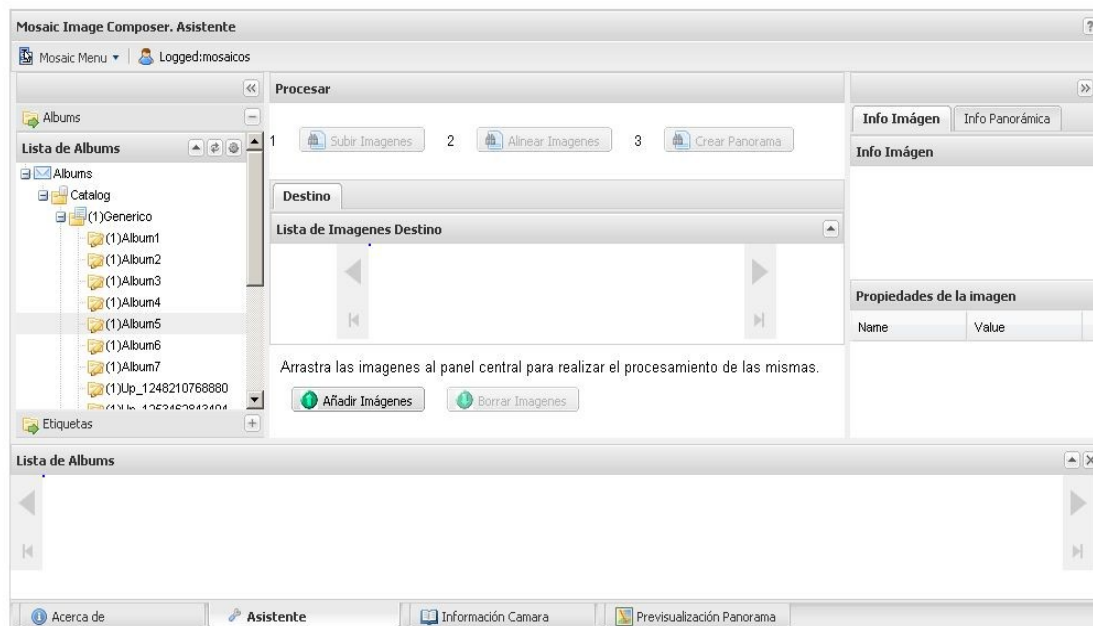


Figura 45: Pestaña Asistente

- Panel Izquierdo.
- Panel Derecho.
- Panel Central.

6.2.2.2.1 Panel Superior

La zona superior de la pantalla permite al usuario realizar varias operaciones como accesos directos. Estas operaciones son las siguientes:

- Activar la consola
- Desactivar la consola
- Acceso a la configuración del proceso de alineación.
- Ver la ayuda online sobre el proceso de alineación.



Figura 46: Panel Menú

La consola permite durante el proceso de alineación ir conociendo el resultado del mismo, permitiendo ver todos los comandos que se van ejecutando para generar el mosaico así como sus resultados.

La consola se activa automáticamente durante el proceso de carga y alineación de imágenes y permite ver la operaciones que se van realizando. La consola se puede ver como una herramienta de visualización de log de la aplicación destinada a analizar cualquier tipo de problema que se pueda producir durante el proceso de alineación.

Al terminar cada fase del proceso la consola se cierra automáticamente por lo que la opción de menú permite mostrarla de nuevo para poder analizar más cómodamente los resultados mostrados en dicha consola.

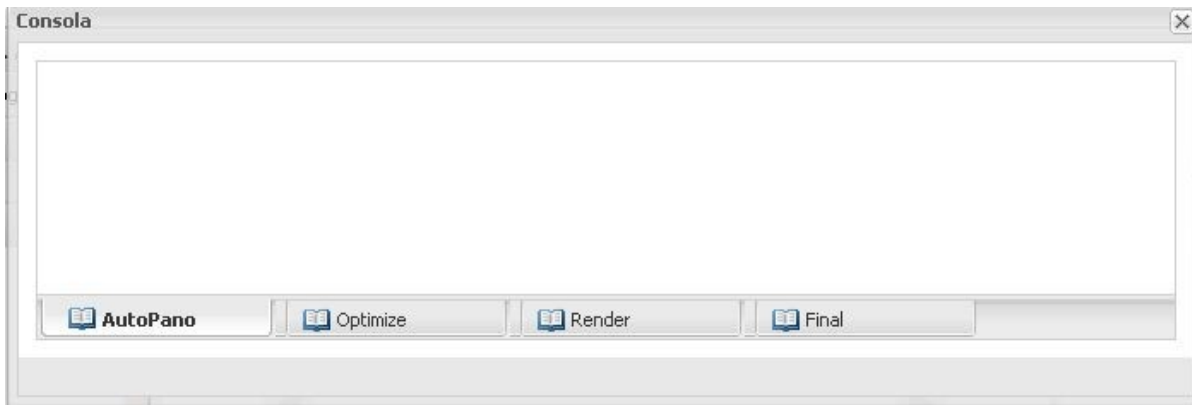


Figura 47: Pestaña Asistente Consola

De la misma forma en esta misma barra de menú aparece el usuario actualmente conectado en el sistema permitiendo conocer en todo momento el perfil del usuario conectado.

En la opción de menú el usuario también tiene la posibilidad de acceder a parámetros específicos de configuración del proceso de alineación. Pulsando sobre esta opción el usuario dispone de una ventana donde puede modificar los parámetros y permitir que el proceso de alineación pueda personalizarse a cada usuario.

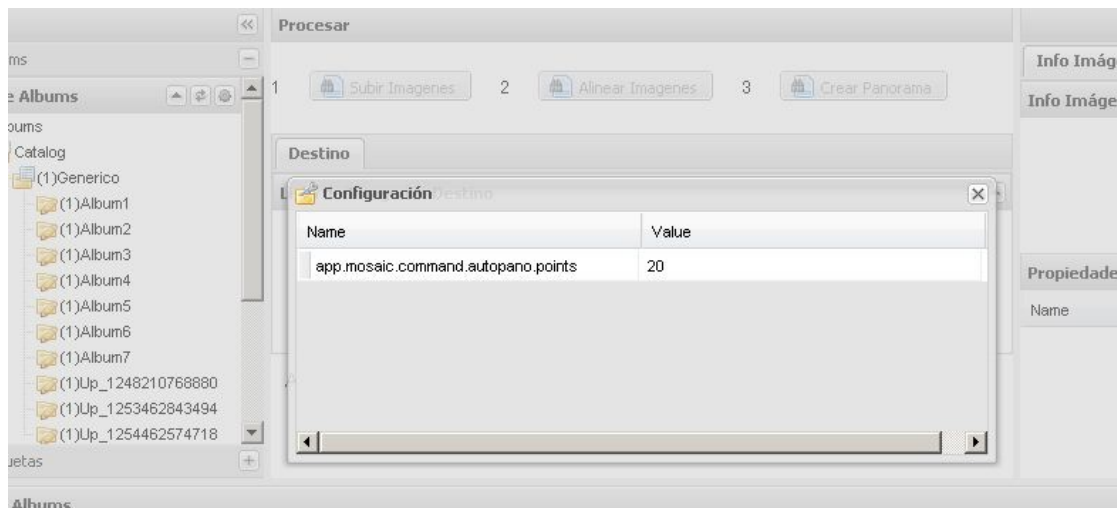



Figura 48: Pestaña Asistente Config

El número de parámetros está limitado actualmente a la posibilidad de poder configurar el número de puntos que se generarán durante el proceso de alineación.

Como última opción el usuario puede acceder como en todas la pantallas a la opción de ayuda en línea sobre las operaciones que se pueden realizar en esta pantalla, para ello deberá pulsar sobre el botón .

6.2.2.2.2 Panel Izquierdo

El panel izquierdo de la pestaña de asistente permite gestionar los contenidos a procesar en la plataforma de generación de mosaicos. A través de esta opción el usuario puede gestionar básicamente dos tipos de contenidos.

- Albums: El concepto de álbum en la plataforma es un conjunto de imágenes con una serie de características que permiten que puedan ser alineadas para conseguir una imagen mayor, fusión de las mismas (mosaico).
- Etiquetas: Las etiquetas permite agrupar álbumes que disponen de ciertas características comunes para que los usuarios puedan tenerlos agrupados bajo ese criterio.

Vista Albums	Vista Etiquetas
--------------	-----------------

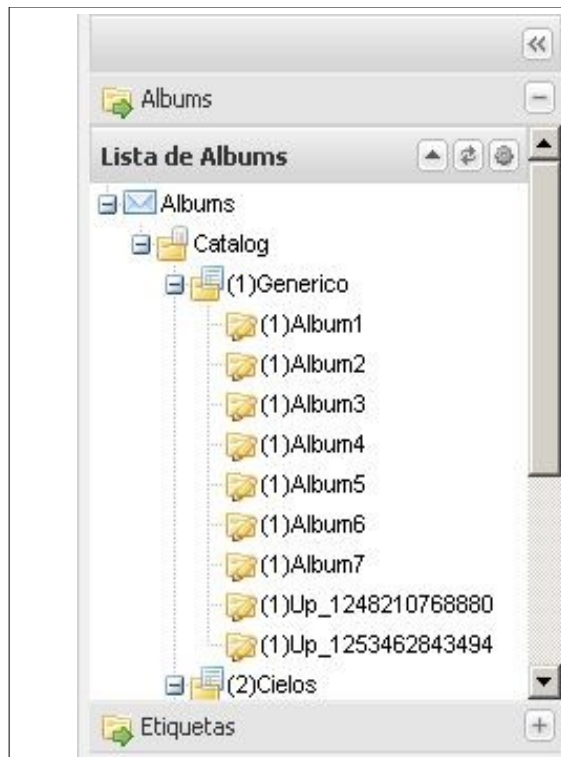




Figura 49: Listado de Albums



Figura 50: Listado de Etiquetas

En este panel los usuarios pueden intercambiar entre la vista de álbumes y la vista de etiquetas simplemente pulsando sobre los botones habilitados para ello.

De la misma forma en este panel disponen de un botón  que permite replegar el panel hacia la derecha permitiendo que el panel central de visualización sea mayor para una mejor visión de las imágenes seleccionadas por el usuario.

Este panel también dispone de dos botones  que permiten el primero realizar una recarga bien de las etiquetas o de los álbumes descargando de nuevo todos los datos del servidor, mientras que el segundo permite realizar una ordenación de las etiquetas o los álbumes de forma alfanumérica con el objetivo de que el usuario pueda visualizar los álbumes en un orden distinto del predeterminado.

En los siguientes párrafos se mostrarán las características más importantes de la gestión que se puede realizar con álbumes y etiquetas.

- **Gestión de álbumes**

El panel permite visualizar el contenido de los distintos álbumes de imágenes del usuario agrupados por las distintas categorías. La estructura de este panel es una estructura arbórea donde el nodo raíz es el catálogo de álbumes.

Como segundo nivel el usuario puede ver las categorías (etiquetas) disponibles para los álbumes del usuario (únicamente aparecen aquellas etiquetas asignadas a los álbumes del usuario permaneciendo el resto ocultas).

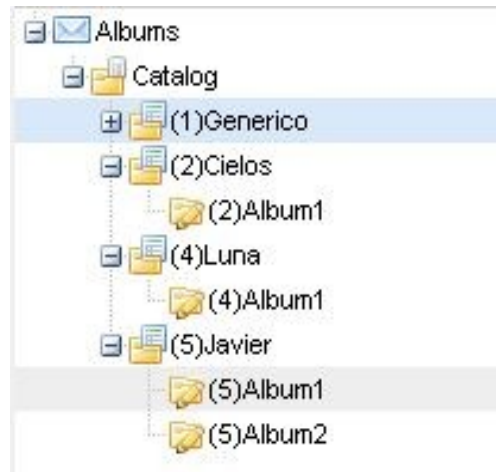


Figura 51: Árbol de Álbumes

Como tercer nivel el usuario puede ver los álbumes dentro de cada etiqueta.

El usuario a través de este panel puede realizar las siguientes operaciones. Estas operaciones dependen del nodo sobre el que se pulse con el botón derecho del ratón.

- Nodo Álbum Genérico.
 - Los álbumes sólo se pueden añadir al nodo genérico para posteriormente añadirle una etiqueta permite catalogarlo en otra categoría. Pulsando con el botón derecho del ratón sobre este nodo al usuario se le presenta un menú que permite seleccionar la opción de generar un nuevo álbum.
 - Al seleccionar la opción de crear un nuevo álbum el sistema introduce en la estructura de árbol como último elemento el nuevo álbum creado.
 - El álbum tiene un nombre generado aleatoriamente bajo el siguiente formato "(1)Up_Fecha". El usuario una vez creado el álbum puede cambiar el nombre simplemente seleccionado el álbum y pulsando en el mismo una vez con el ratón. El sistema resalta el álbum y permite que el usuario pueda realizar la modificación.

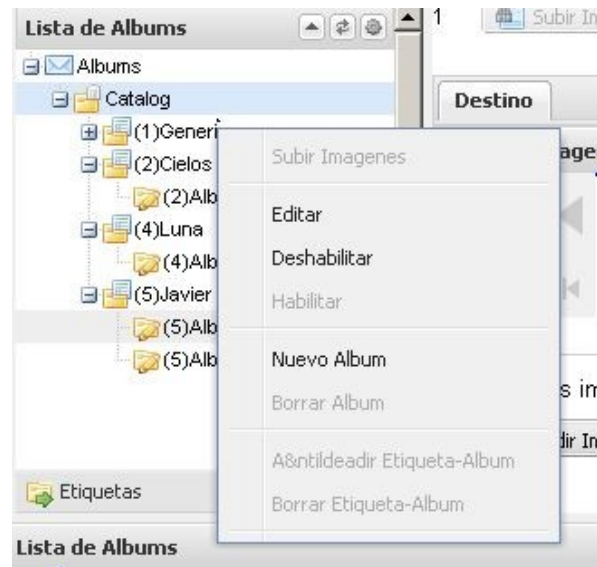


Figura 52: Nuevo Álbum

En la Figura 53 se puede ver como el usuario puede seleccionar un album.

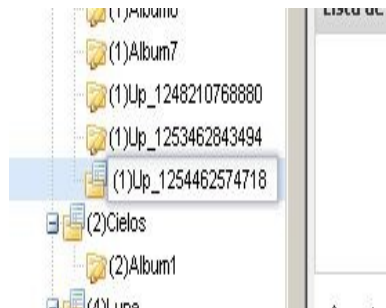


Figura 53: Selección de un Álbum

- Nodo Álbum.

- Las opciones disponibles seleccionando directamente uno de los álbumes previamente creados son más extensas que las anteriores. Entre las opciones accesibles se disponen de las siguientes:
 - 1 Visualización de imágenes: Pulsando con el botón izquierdo del ratón sobre el álbum las imágenes contenidas en el mismo se muestran automáticamente en el panel inferior de la pestaña de asistente.
 - 2 Las siguientes operaciones son accesibles mediante un menú que se presenta al

usuario al pulsar con el botón derecho del ratón

- Renombrar un álbum.
- Borrar un álbum.
- Habilitar/Deshabilitar un álbum.
- Subir imágenes a un álbum.
 - A través de esta opción el usuario puede subir imágenes locales al álbum que ha creado en el servidor. El sistema presenta al usuario un diálogo donde debe seleccionar las imágenes que quiere subir.

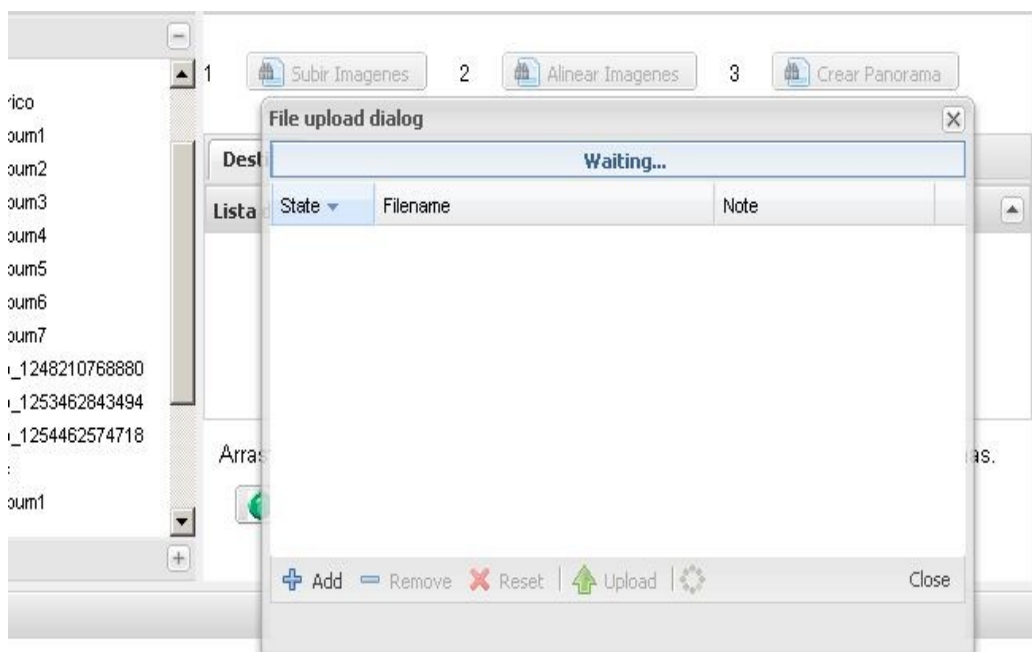


Figura 54: Subida de imágenes (Paso1)

- El usuario únicamente puede seleccionar imágenes de una en una, las cuales se van almacenando en el panel de subida.

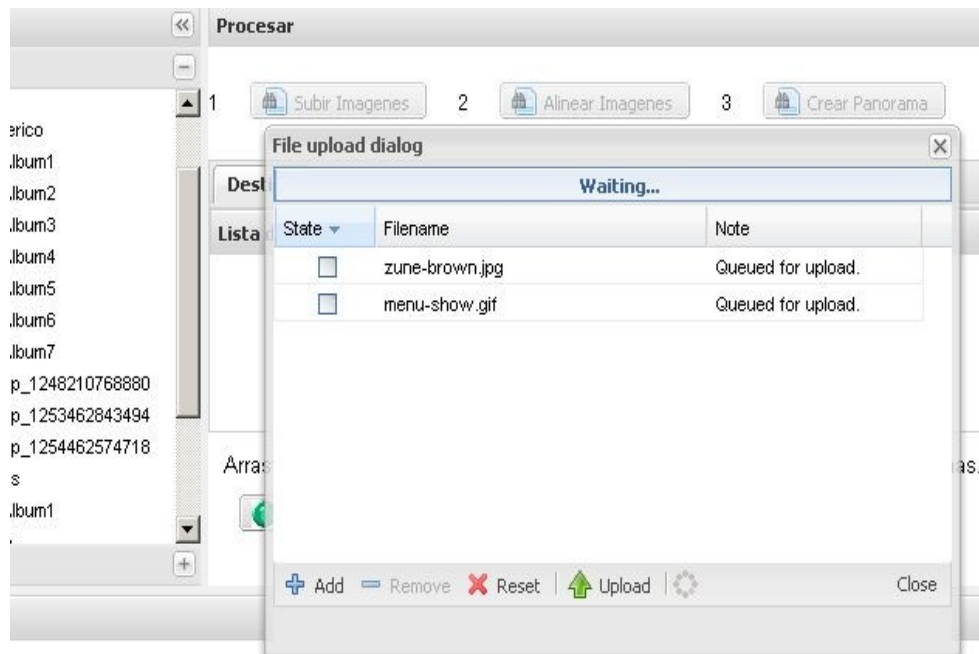


Figura 55: Subida de imágenes (Paso2)

- Una vez seleccionadas todas las imágenes el usuario pulsa el botón Upload y las imágenes son subidas al servidor para formar parte del álbum creado por el usuario.
- ➔ Añadir/Borrar etiquetas a un álbum.
 - A través de esta opción el sistema permite que el usuario asocie al álbum seleccionado una etiqueta.
 - Al pulsar en esta opción al usuario se le presenta una ventana con el listado de etiquetas disponibles. El usuario deberá seleccionar la etiqueta y arrastrarla al árbol seleccionado en pantalla

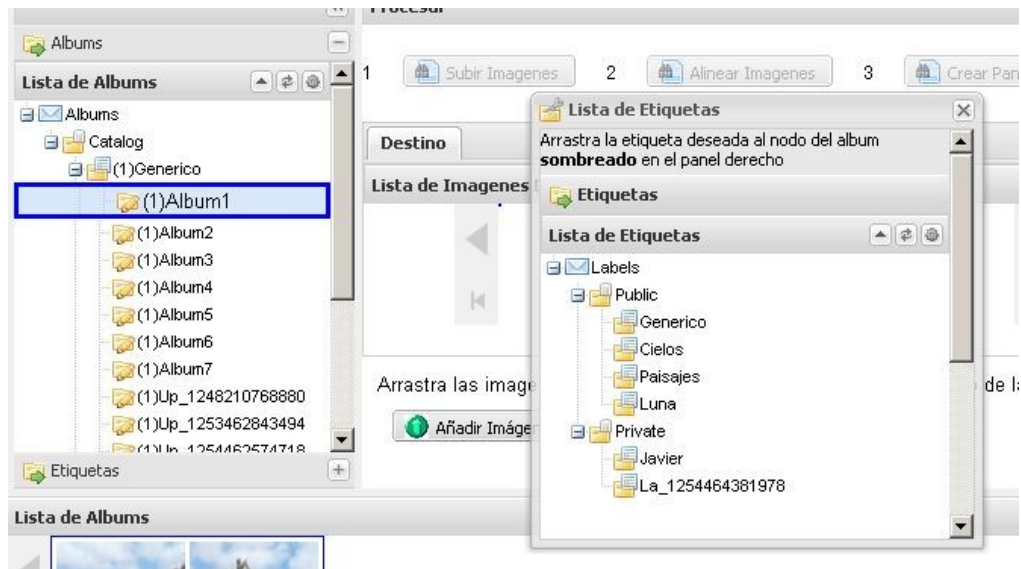


Figura 56: Añadir Etiqueta (Paso1)

Una vez seleccionada la etiqueta, el usuario simplemente la desplaza hacia el álbum deseado. Como se puede ver en la siguiente ilustración el usuario desplaza la etiqueta “Etiqueta1” al álbum “Album1” el cual está resaltado en color verde.

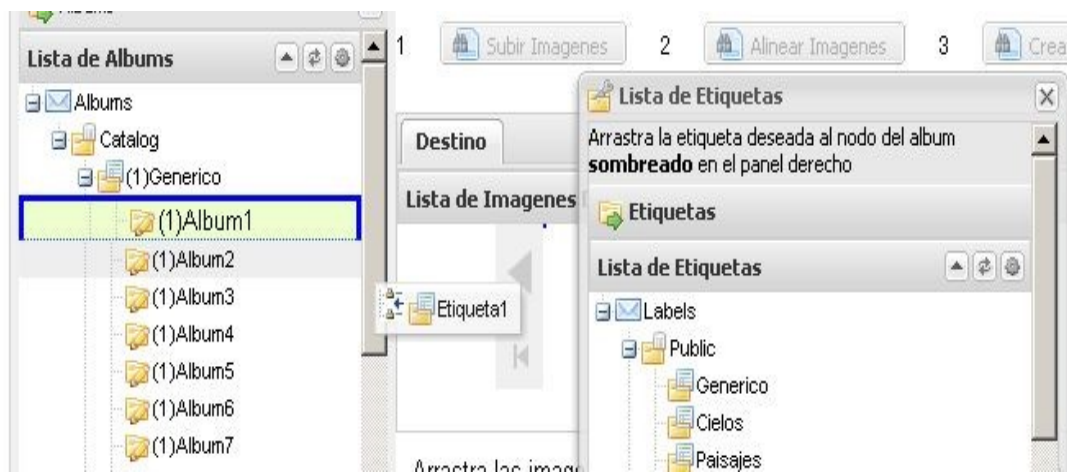


Figura 57: Añadir Etiqueta Paso2

- Gestión de etiquetas

A través de este panel el usuario puede gestionar las etiquetas públicas (disponibles para todos los usuarios) y privadas (personalizadas para un usuario en particular).

La estructura de esta zona es de la misma forma que la de los álbumes arbórea en este caso con sólo dos niveles.

El primer nivel permite gestionar las etiquetas públicas o privadas mientras que el segundo nivel permite gestionar ya directamente las etiquetas.

El usuario a través de este panel puede realizar las siguientes operaciones. Estas operaciones de la misma forma que en el caso de los álbumes dependen del nodo sobre el que se pulse con el botón derecho del ratón.

- Nodo Public.
 - Al ser el nodo público un nodo gestionado a nivel de administración no es posible generar nuevas etiquetas. La creación de nuevas etiquetas para todos los usuario solo está disponible a nivel administrador.
- Nodo Private.
 - A través de este nodo los usuarios pueden crear nuevas etiquetas para asociar a sus contenidos. Simplemente pulsando con el botón derecho del ratón al usuario se le presenta un menú donde puede seleccionar las siguientes operaciones.

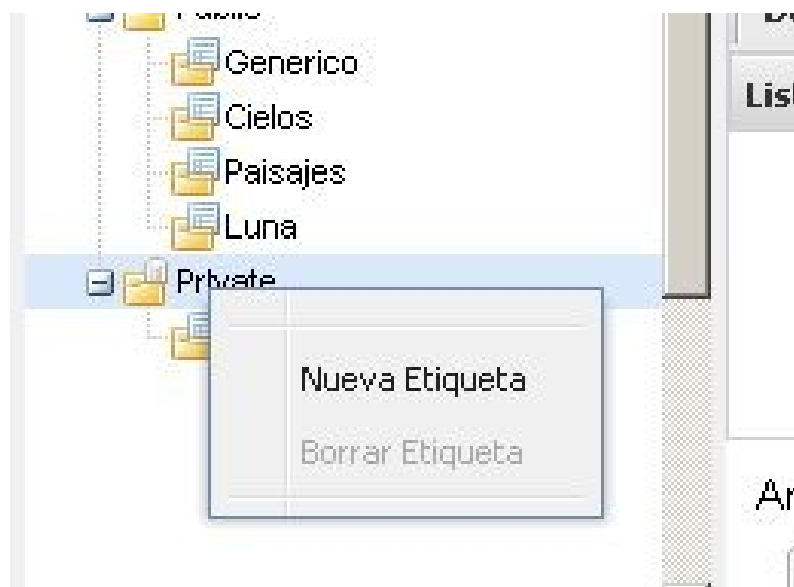


Figura 58: Creación de nueva etiqueta

- Nueva Etiqueta: Mediante esta opción el usuario genera una nueva etiqueta dentro de la categoría de etiquetas privadas.
- Al seleccionar la opción de crear una nueva etiqueta, el sistema introduce en la

estructura de árbol como último elemento la nueva etiqueta creado. La etiqueta tiene un nombre generado aleatoriamente bajo el siguiente formato “La_Fecha”. El usuario una vez creada la etiqueta puede cambiar el nombre simplemente seleccionándola y pulsando en la misma una vez con el ratón. El sistema resalta la etiqueta y permite que el usuario pueda realizar la modificación

- Nodo Etiqueta.
 - Las opciones disponibles sobre el nodo etiqueta previamente creado son accesibles mediante la utilización del botón derecho del ratón. Al pulsar sobre la etiqueta y una vez presentado el menú contextual, el usuario puede realizar la siguiente operaciones.
 - Renombrar una etiqueta: Mediante esta opción el usuario puede renombrar la etiqueta seleccionada por el usuario.
 - Borrar una etiqueta. Es importante reseñar que borrar una etiqueta supone que los álbumes que tuvieran asignados esa etiqueta perderán dicha asignación.

6.2.2.2.3 Panel Derecho

A través de este panel el usuario tiene acceso a dos subpaneles de la aplicación.

- Información de la imagen
- Información de la panorámica.



Figura 59: Panel Derecho

Panel de Información de Imagen.

Este panel permite conocer la información (imagen + características) de una imagen seleccionada por el usuario. Una vez que el usuario ha seleccionado un álbum, las imágenes disponibles se despliegan sobre el panel inferior de la ventana. El usuario situando el ratón sobre cualquiera de las imágenes provoca que la información de la misma se muestre sobre este panel.

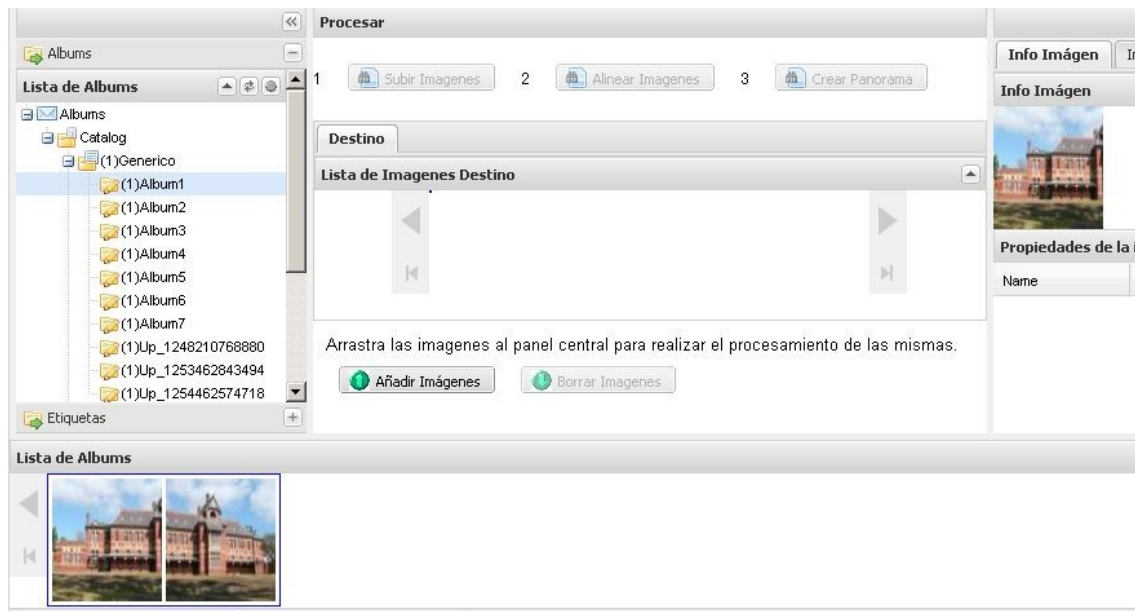


Figura 60: Información de imagen

En la imagen se puede ver el que el usuario tiene el puntero del ratón sobre una imagen y esta y sus características se muestran en el panel derecho.

Entre las características que se muestran en el panel derecho se muestran las siguientes.



Figura 61: Información de la imagen

- Nombre de la imagen.
- Factor de recorte (cropFactor).
- Distancia Focal (FocalLength).
- Vista Horizontal de la imagen (HFOV).
- Proyección (Projection).
- Nombre.
- Exposición.

Panel de Información de Panorámica.

La información de este panel permite visualizar un *thumbnail* de la imagen panorámica generada así como la posibilidad de descargar la misma al PC del usuario pulsando sobre el botón descargar.

Es importante reseñar en este punto que la imagen panorámica generada se almacena asociada al álbum de forma permanente y permite que en todo momento, incluso si el usuario cierra el navegador y lo vuelve a abrir, pueda visualizar o descargar la panorámica. Esto va a permitir que los usuarios puedan visualizar las panorámicas en todo momento simplemente seleccionando el álbum deseado.

En las siguientes ilustraciones se muestra la visualización del panel cuando el álbum dispone de panorámica Figura 62 y cuando no dispone Figura 63.

Este panel mostrará la imagen generada en dos supuestos:

- Cuando el usuario genere en el momento la panorámica sobre el álbum seleccionado.
- Cuando el álbum ya dispone de una panorámica generada en un proceso de generación previo como se ha comentado anteriormente.

6.2.2.2.4 Panel Central

- El panel central de la pestaña de asistente permite realizar el procesamiento de las imágenes con el objetivo final de generar la panorámica Figura 64




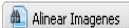

Álbum con panorámica	Álbum sin panorámica
 <p><i>Figura 62: Álbum con panorámica</i></p>	 <p><i>Figura 63: Álbum sin panorámica</i></p>



Figura 64: Panel central

El panel central está formado por dos subpaneles, el inferior permite visualizar las imágenes que van a ser objetivo de la panorámica mientras que el superior dispone de tres botones que permiten realizar las tres fases del proceso de alineación.

- Carga de imágenes. 
 - La carga de imágenes es el paso previo que se realiza antes de poder generar la panorámica. En este proceso las imágenes se suben al servidor y se comprueban ciertas características de las mismas para verificar que pueden ser objetivo del proceso de alineación.
- Proceso de alineación. 
 - El proceso de alineación consiste en una vez subidas las imágenes al servidor obtener una previsualización del mosaico. Para realizar esta operación el sistema inicialmente intenta obtener una serie de características comunes en la fotografía utilizando algoritmos de reconocimiento de imágenes. Si el sistema detecta elementos comunes el proceso de alineación puede continuar con el proceso de rustico (cosido) de la imagen para generar la panorámica.
- Proceso de creación de panorama. 
 - Una vez que se ha realizado la previsualización del mosaico el último paso consiste en generar la imagen definitiva para lo cual se realizan un proceso de optimización sobre la imagen generada en el paso anterior. En muchos casos la imagen generada en este proceso

no es muy distinta a la previsualizada en el paso anterior aunque este último proceso utiliza programas específicos para generar la panorámica.

El usuario debe desplazar las imágenes del panel inferior al superior bien mediante la utilización del soporte *drag-drop* o bien mediante el uso de los botones habilitados al efecto que ofrecen una mayor velocidad a la hora de realizar el movimiento de imágenes entre los distintos paneles.

6.2.2.2.5 Panel Inferior





El panel inferior de la pestaña de asistente permite visualizar las imágenes del álbum seleccionado por el usuario.


Estás imágenes se muestran en forma de presentación con una serie de botones que permiten desplazar las imágenes a la izquierda o a la derecha del panel si las imágenes no caen dentro del área de visualización del mismo. Esta característica es muy importante dado que en una composición el número de imágenes que forman el mosaico puede ser muy elevado.



Figura 65: Panel inferior

El panel dispone de los siguientes botones para realizar las siguientes operaciones.

Botón	Operación
	Al pulsar este botón el usuario desplaza las imágenes hacia la izquierda permitiendo visualizar las imágenes que están más hacia la derecha del panel. Este botón permanece desactivado si no existen más imágenes en la zona de la derecha del panel
	Al pulsar este botón el usuario desplaza las imágenes hacia la derecha permitiendo visualizar las imágenes que están más hacia la izquierda del panel. Este botón permanece desactivado si no existen más imágenes en la zona de la izquierda del panel. Por defecto cuando se presentan las imágenes el botón permanece deshabilitado al mostrar el panel la primera imagen alineada a la izquierda
	Al pulsar este botón el usuario desplaza las imágenes hacia el final del listado de imágenes permitiendo visualizar las últimas imágenes de la presentación
	Al pulsar este botón el usuario desplaza las imágenes hacia el

	comienzo del listado de imágenes permitiendo visualizar las primeras imágenes de la presentación. Por defecto cuando se presentan las imágenes el botón permanece deshabilitado al mostrar el panel la primera imagen alineada a la izquierda
	Este botón permite eliminar del panel todas las imágenes seleccionadas por el usuario. Las imágenes que se eliminan del panel inferior no afectan al número de imágenes que previamente se han seleccionado para generar el mosaico.

El panel inferior también dispone de una característica muy interesante a nivel de usuario y es el soporte de drag-drop sobre las imágenes visualizadas en dicho panel.

El usuario seleccionando una imagen o un conjunto de imágenes con el ratón y arrastrándolas con el mismo puede situar dichas imágenes en el panel de procesamiento situado en la zona central de la pantalla (El panel central dispone también de dicha característica). Esto permite de una forma muy cómoda para el usuario añadir (del panel inferior al panel central) y eliminar (del panel central al panel inferior) imágenes para el procesamiento de las mismas.

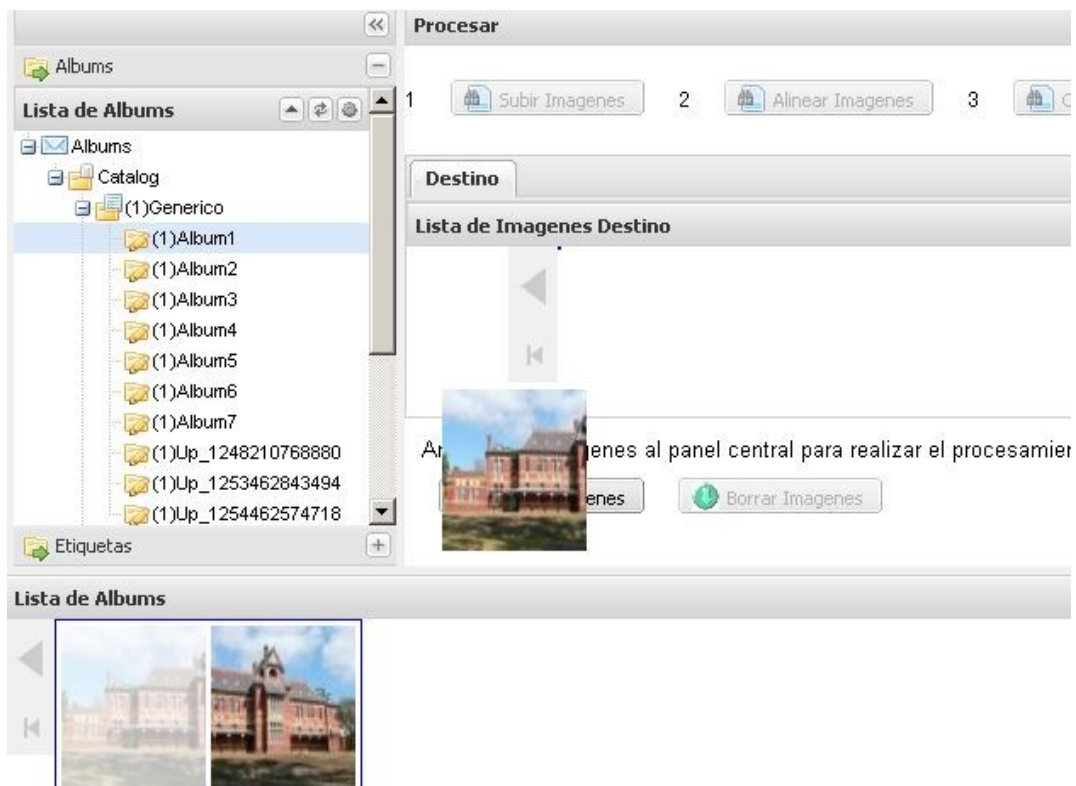


Figura 66: Drag & Drop

En la ilustración se puede ver como la imagen se desplaza del panel inferior al panel central y se marca como desactivada en el panel inferior.

6.2.3 Pestaña “Información Cámara”

Esta pestaña permite personalizar la información de las imágenes almacenadas. Una visión general de esta pestaña se ofrece en la siguiente ilustración.

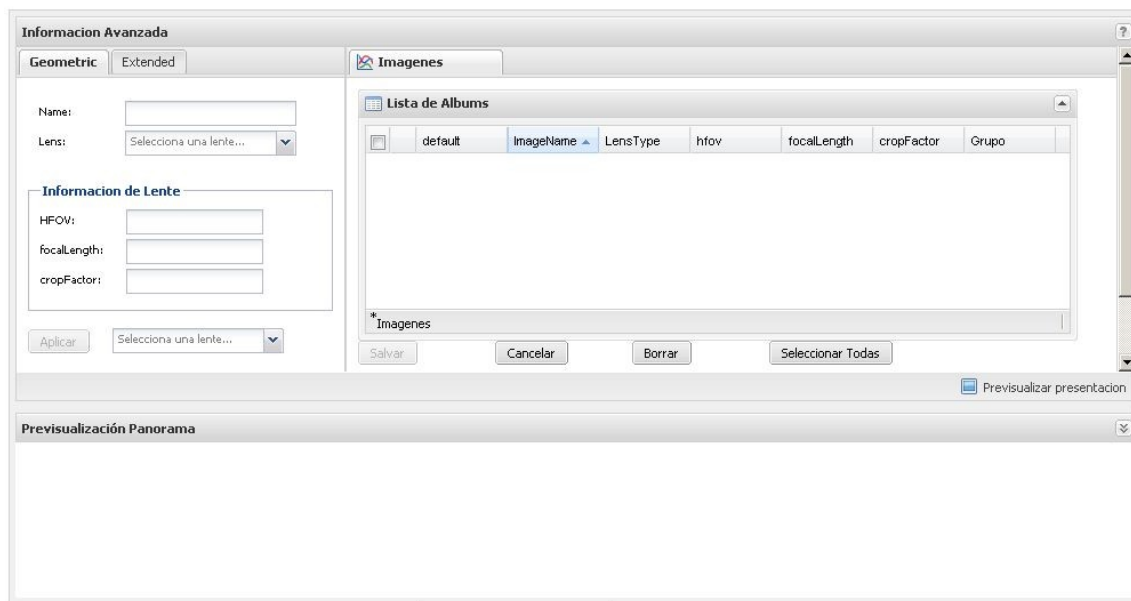


Figura 67: Pestaña Camera

La pantalla principal está organizada siguiendo la siguiente estructura en forma de paneles.

- Panel Izquierdo.
- Panel Derecho.
- Panel Inferior.

6.2.3.1 Panel Izquierdo

El panel izquierdo de la pestaña de información de cámara permite modificar mediante una serie de formularios personalizados la información geométrica y optométrica de las imágenes seleccionadas por el usuario para el proceso de alineación.

La información que se necesita para el proceso de alineación es variada y se obtiene fundamentalmente de la cámara y más concretamente de la lente o grupos de lentes con los que se realiza la imagen.

La mayoría de las cámaras modernas disponen de la posibilidad de almacenar una serie de metadatos sobre las fotos que se generan, esta información se conoce con el nombre de Exif.

EXIF es una acrónimo de “Exchangeable Image File Format”, básicamente es una estandarización

de la información sobre el formato de la imagen. El exif, que se encuentra dentro del mismo archivo, a modo de metadatos, contiene toda la información acerca de la imagen: su estructura, tamaño, su codificación, fecha de creación, entre muchísimos otros atributos. Esta estandarización permite que los distintos software y scripts puedan leer e interpretar esta información para la manipulación de la imagen.

Al cargar las imágenes en la herramienta el sistema lee la información del exif que viene en la imagen y la presenta en el formulario con el objetivo de que el usuario pueda visualizar dicha información y potencialmente cambiarla.

No obstante existe una posibilidad bastante alta de que las imágenes no dispongan de esa información ya que muchos programas de tratamiento de imágenes, con el fin de agilizar el tamaño de las mismas, eliminan toda la información de metadatos de la imagen.

A través de este panel el usuario puede modificar los valores del exif que afectan potencialmente al proceso de alineación. Entre estos valores tenemos los siguientes.

Figura 68: Panel de Información Geométrica

- **HFOV:** El campo de visión (expresado en metros, pies, pulgadas, etc.) es la porción de la imagen visible a través de un objetivo acoplado a una cámara. Este factor se ve afectado por el valor de la distancia focal por lo que modificando la distancia focal se modifica el factor HFOV.
- **Distancia Focal (focalLength):** La distancia focal o longitud focal de una lente es la distancia

entre el centro óptico de la lente y el foco (o punto focal). Un valor bajo de distancia focal provoca un ángulo de visión amplio, mientras que, un valor alto implica un ángulo de visión reducido. Este factor se ve afectado por el valor de HFOV por lo que modificando el HFOV se modifica la distancia focal.

- **Factor de recorte (cropFactor):** Erróneamente llamado factor de multiplicación, indica la proporción entre el tamaño del sensor en algunas cámaras reflex digitales y el tamaño original de las de película de 35mm. El cálculo exacto del factor de recorte se obtiene dividiendo la diagonal del formato "Full Frame" por la diagonal del sensor correspondiente.
- **Proyección (Tipo de Lente):** El tipo de lente indica las características de la lente con la que se ha tomado la fotografía. Esta lente puede tomar distintos valores (normal, full-frame, fish-eye (ojo de pez, panorámica, etc.).

Como se ha comentado anteriormente los valores de óptica de la imagen afectan unos a otros por lo que el sistema realiza los cálculos automáticamente para que los valores de cada uno de los campos sean correctamente calculados.

Tip: Es complicado si no se conoce la información de la lente obtener los valores correctos por lo que lo aconsejable es usar un valor estimado. Como guía las lentes normales de una cámara de 35mm disponen de una distancia focal de 50mm y un valor de HFOV de unos 40 grados. Una cámara de gran angular de 28mm dispone de un HFOV de 64 grados.

Esta información para personas que no son expertas en lentes y cámaras digitales es bastante complicado de introducir si la imagen no dispone de metadatos (exif), con el objetivo de que los usuarios puedan realizar composiciones óptimas el sistema dispone de la posibilidad de poder aplicar lentes a las imágenes seleccionadas por el usuario.

Una lente básicamente es un conjunto de valores predeterminados para los parámetros anteriormente comentados. Debido a que la aplicación está especialmente diseñada para la composición de imágenes del espacio el sistema dispone de una serie de lentes predefinidas que ayudan a que el usuario no tenga que conocer la información de las imágenes.

HFOV: 22

focalLength: 4047044355049122,00

cropFactor: 22

Aplicar

Telescope-1

Telescope-2

Telescope-3

Previsualización Panorama

Figura 69: Lentes disponibles

El usuario tras seleccionar la imagen en el panel central a la que quiere aplicar la lente , selecciona la lente deseada y únicamente debe pulsar el botón aplicar para que los valores de la lente se copien en los campos del formulario y se apliquen sobre la imagen seleccionada por el usuario.



Figura 70: Aplicación de lentes en imágenes

6.2.3.2 Panel Central

El panel central permite que el usuario pueda visualizar el listado de imágenes seleccionadas por el usuario en la pestaña de asistente así como sus características (metadatos).

Las imágenes se presentan en un tabla en el que por cada fila se presenta el nombre de la imagen, el tipo de lente (proyección), el hfov, la distancia focal y el factor de recorte así como un parámetro que luego cobrará importancia a la hora de seleccionar las imágenes, **el grupo**.

Lista de Albums								
<input type="checkbox"/>			ImageName	LensType	hfov	focalLength	cropFactor	Grupo
Grupo: GRUPO0 (19 Items)								
<input checked="" type="checkbox"/>	1	false	luna-1.jpg	0	29,30000	66,20434	1,00000	GRUPO0
<input type="checkbox"/>	2	false	luna-10.jpg	0	29,30000	66,20434	1,00000	GRUPO0
<input type="checkbox"/>	3	false	luna-11.jpg	0	29,30000	66,20434	1,00000	GRUPO0
<input type="checkbox"/>	4	false	luna-12.jpg	0	29,30000	66,20434	1,00000	GRUPO0
* Imágenes								

Figura 71: Lista de imágenes

Durante el proceso de alineación de las imágenes el sistema detecta las imágenes que tienen

características similares y la agrupa de tal forma que posteriormente se puede generar la imagen mosaico. En el proceso cada **grupo** que se genera se identifica con un nombre de tal forma que en la tabla se generan agrupaciones de imágenes.

Si existen imágenes no relacionadas el sistema muestra un mensaje de error para que el usuario lo subsane. Este mensaje se puede ver en la figura Figura 72



Figura 72: Grupos no conectados

El sistema como muestra la figura Figura 73 indica al usuario que imágenes están dentro de cada grupo.

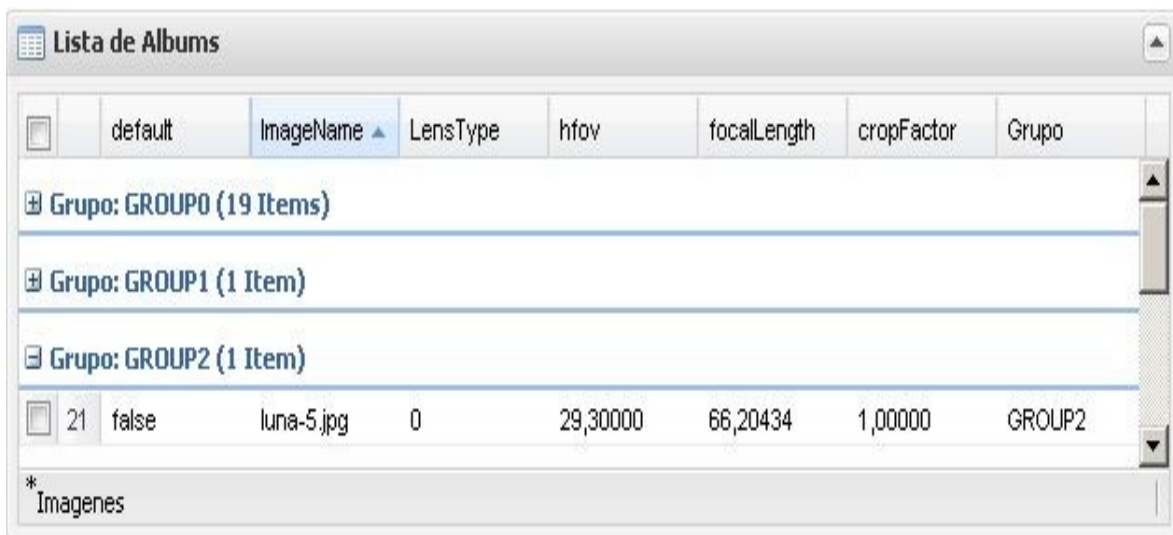






Figura 73: Agrupación de imágenes

En este panel el usuario dispone de un primer campo ☒ que permite que el usuario pueda seleccionar la fila correspondiente a la imagen deseada por el usuario.

Debajo de la tabla de imágenes el usuario dispone de cuatro botones que le permiten aplicar o eliminar la información deseada. Entre las operaciones que se pueden realizar están las siguientes:

-
- **Salvar:** Pulsando el botón  los cambios realizados por el usuario en el panel izquierdo (óptica) se aplican a las imágenes seleccionadas por el usuario.
 - **Cancelar:** Pulsando el botón  los cambios realizados por el usuario en el panel izquierdo se descartan.
 - **Borrar:** Pulsando el botón  el usuario elimina de la composición la imagen o grupo de imágenes seleccionadas por el usuario.
 - **Seleccionar Todas:** Pulsando el botón  el usuario selecciona todas las imágenes disponibles en el panel central de imágenes, permitiendo bien agilizar el proceso de modificación de características en bloque o borrar todas las imágenes.

6.2.3.3 Panel Inferior

El panel inferior permite que el usuario pueda visualizar una composición animada de las imágenes seleccionada para el procesamiento de la panorámica. Esta presentación es únicamente descriptiva y no puede modificar el estado de la alineación

En este panel inferior el usuario dispone del botón de previsualización que permite presentar sobre el panel inferior la composición de las imágenes. En la Figura 74 se puede ver este panel de visualización.

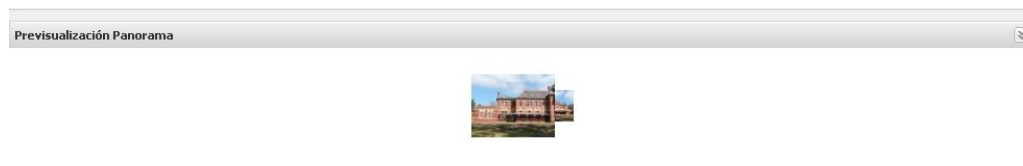


Figura 74: Previsualización de imágenes

Las imágenes son presentadas con una de las imágenes por delante de las otras mientras que el resto aparece en un segundo plano. Pulsando sobre la imagen que está en segundo plano ésta se desplaza y se coloca en primer plano pasando esta última a un segundo plano.



Figura 75: Previsualización de imágenes. Desplazamiento

6.2.4 Pestaña “Previsualización Panorama”

Esta pestaña permite visualizar el resultado de la composición generada. Una visión general de esta pestaña y su estructura se ofrece en la siguiente ilustración.

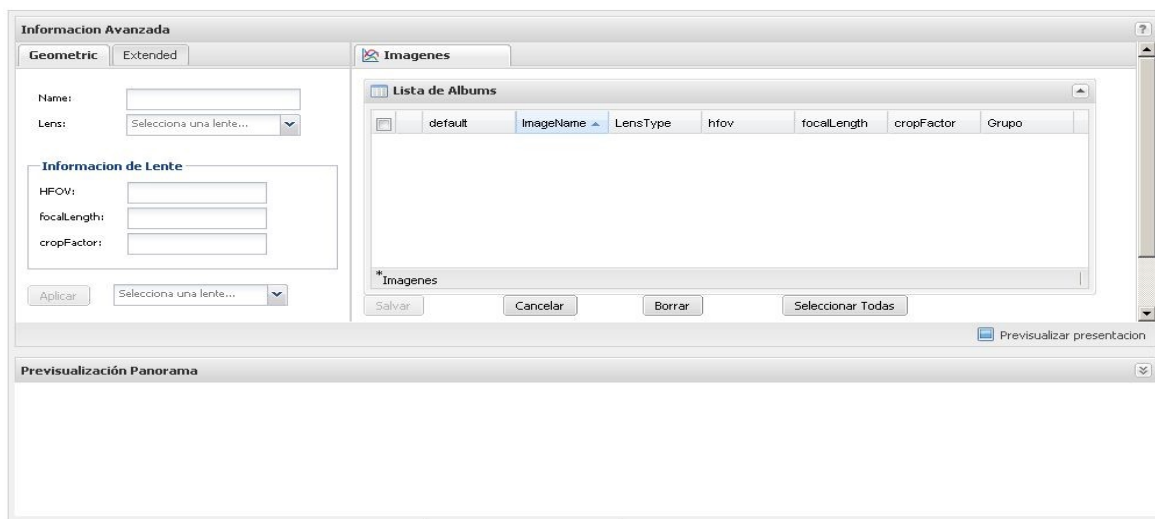


Figura 76: Pestaña Camera

La pantalla principal está dividida en tres pestañas en la zona inferior del panel.

- Pestaña de información.

-
- Pestaña de previsualización de la alineación
 - Pestaña de previsualización final



Figura 77: Pestañas Previsualización

6.2.4.1 Pestaña Información

En esta pestaña el usuario puede ver los datos básicos del proceso de alineación. Estos datos hacen referencia a las distintas URL's que permiten interaccionar con el resultado final del proceso de alineación.

- **URL Preview:** La URL de previsualización permite acceder a la aplicación de visualización de la imagen panorámica. Introduciendo la URL en un navegador el usuario puede visualizar la imagen y navegar por la misma a través de una serie de controles.
 - Este mismo comportamiento se puede realizar directamente desde la pestaña de previsualización integrada dentro de esta misma herramienta aunque la URL permite poder incluir esa URL en otro portal y visualizarla.
- **URL Descarga:** La URL de descarga permite que el usuario pueda descargar la imagen de mosaico generada. Introduciendo la URL en un navegador el sistema le presenta un diálogo para que guarde la imagen en su PC.

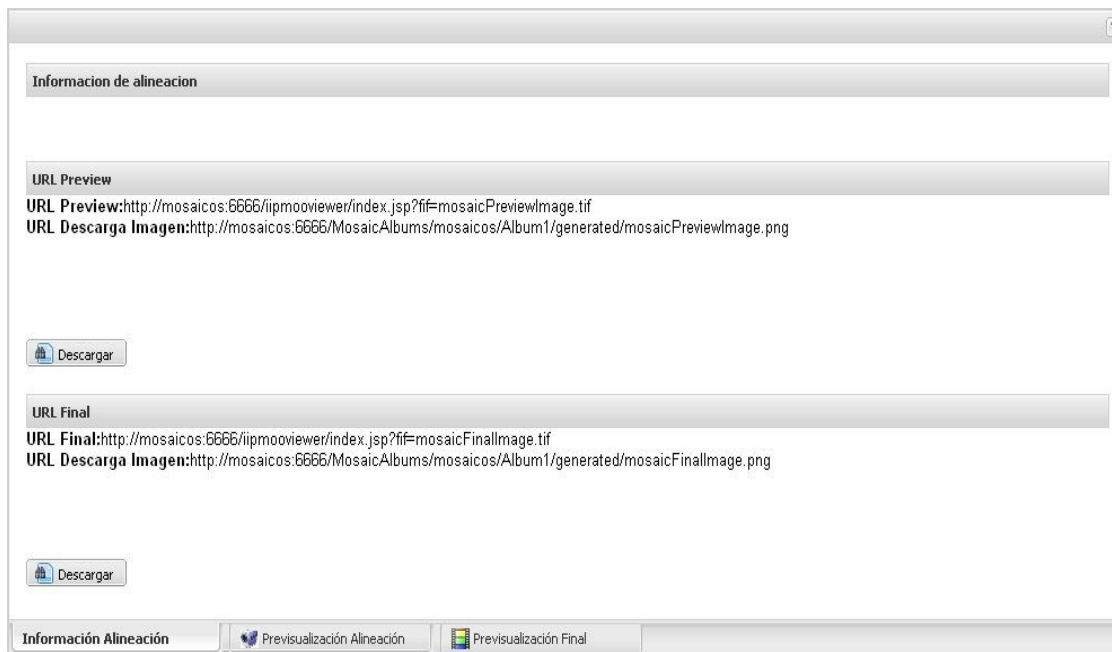


Figura 78: Información alineación

Esta información se replica para la imagen de previsualización que se genera en el segundo paso así como para la imagen que se genera en el último paso.

6.3 Proceso de generación de mosaicos

En este apartado se mostrará al usuario los pasos necesarios para llevar a cabo la composición del mosaico desde el momento en que se suben las imágenes al servidor hasta que el usuario puede visualizar el mosaico desde la propia herramienta.

Esta sencilla guía pretende completar los apartados anteriormente explicados, permitiendo una mejor comprensión del proceso de generación de mosaicos.

El proceso de composición de mosaicos comienza, una vez que el usuario se ha autenticado en el sistema, con la preparación del entorno para la generación del mismo.

6.3.1 PASO1. Preparación del álbum contenedor

Este paso se divide en 5 subpasos que se explican a continuación.

PASO 1.1

El primer paso que debe realizar el usuario es la creación de un álbum para alojar las imágenes.

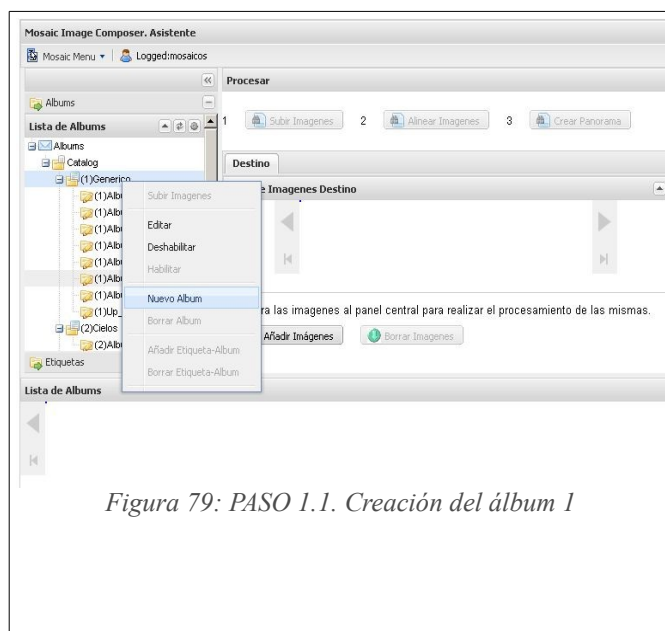


Figura 79: PASO 1.1. Creación del álbum 1

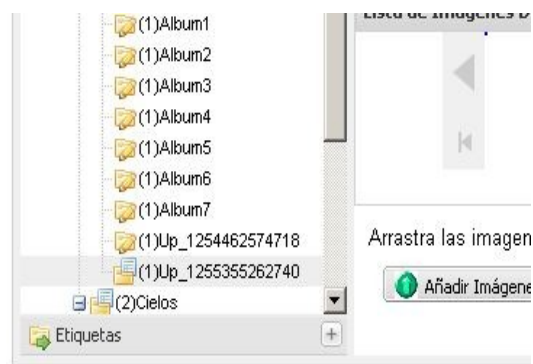


Figura 80: PASO 1.1. Creación del álbum 2

El usuario debe seleccionar en el nodo con nombre **“Genérico”** con el botón derecho del ratón y automáticamente le aparecerá un menú contextual en el que podrá seleccionar la opción de Crear un nuevo Álbum.

Una vez ejecutada la operación el sistema genera un nuevo nodo al final del árbol con un nombre

aleatorio que posteriormente el usuario podrá cambiar. En la ilustración podemos ver que se le asigne el nombre “(1)Up_1255355262740”.

El formato del nombre del álbum es el siguiente:

- (IdCategoria) Identificador interno de la categoría sobre la que se sitúa el nodo. En el ejemplo aparece como (1).
- Up_: Indica que el Álbum se ha creado para realizar upload de imágenes.
- 1255355262740: Nombre del nodo generado de forma aleatoria.

PASO 1.2

El usuario una vez creado el álbum dispone de la posibilidad de modificar el nombre del mismo para adaptarlo al tipo de imágenes que va a albergar.

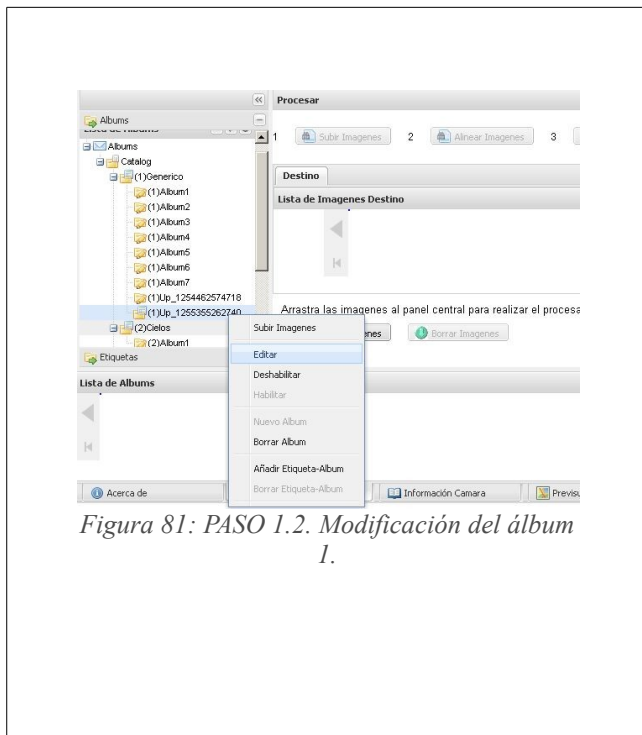


Figura 81: PASO 1.2. Modificación del álbum 1.



Figura 82: PASO 1.2. Modificación del álbum 2.

El usuario debe seleccionar el nodo que desea modificar con el botón derecho del ratón y automáticamente le aparecerá un menú contextual en el que podrá seleccionar la opción de Editar el álbum.

Una vez ejecutada la operación el sistema permite que el usuario pueda introducir el nombre del álbum que desea. En este caso se ha introducido el nombre “**Fotos_Luna**”. El sistema automáticamente añade en el nombre del álbum el identificador de la categoría en la que se sitúa el álbum. En este caso aparecerá como “(1)**Fotos_Luna**”. Es importante remarcar en este punto que el nombre del álbum debe cumplir una serie de características básicas como por ejemplo empezar por letra o número y no

contener caracteres especiales. Los únicos caracteres permitidos son los siguientes (“-”, “_”).

PASO 1.3

La siguiente operación como la comentada anteriormente es opcional para el proceso de generación de mosaicos pero permite que el usuario puede tener catalogada toda la información.

En esta operación el usuario puede asignar una categoría al álbum generado para tener ordenada toda la información.

En este proceso el paso inicial va a ser generar la categoría en el caso de que esta no estuviese creada previamente.

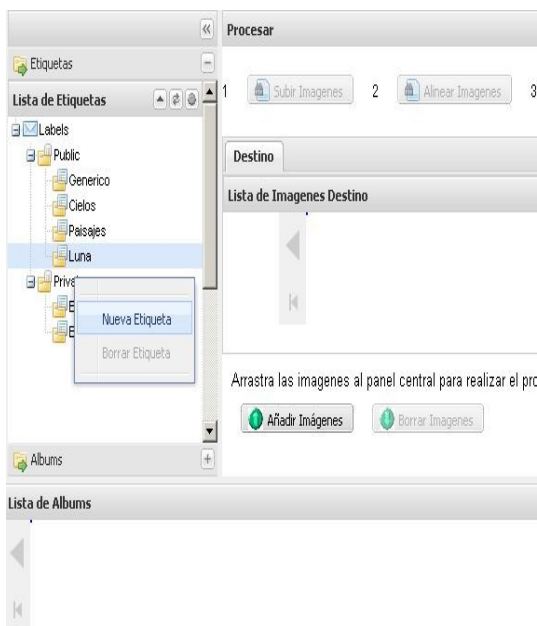


Figura 83: PASO 1.3. Creación de categoría 1

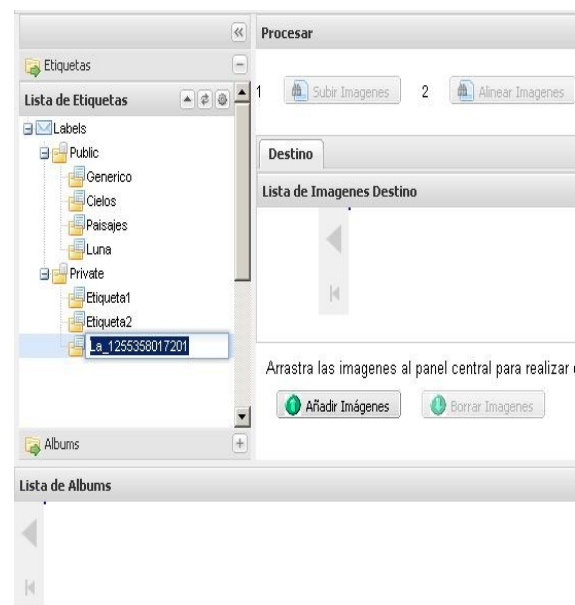


Figura 84: PASO 1.3. Creación de categoría 2

El usuario debe seleccionar en el nodo con nombre **“Private”** dentro de la lista de categorías con el botón derecho del ratón y automáticamente le aparecerá un menú contextual en el que podrá seleccionar la opción de **“Crear Categoría”**. Únicamente se pueden generar categorías dentro del nodo **Private**. Las categorías públicas están predefinidas en el sistema y sólo pueden modificar a nivel administrador.

Una vez ejecutada la operación el sistema genera un nuevo nodo al final del árbol con un nombre aleatorio que posteriormente el usuario podrá cambiar. En la ilustración podemos ver que se le asigne el nombre **“La_1255358017201”**.

El formato del nombre del álbum es el siguiente:

- La_: Indica que el nodo es de tipo Label (Etiqueta).
- 1255358017201: Nombre del nodo generado de forma aleatoria.

PASO 1.4

El usuario una vez creada la categoría dispone de la posibilidad de modificar el nombre de la misma.

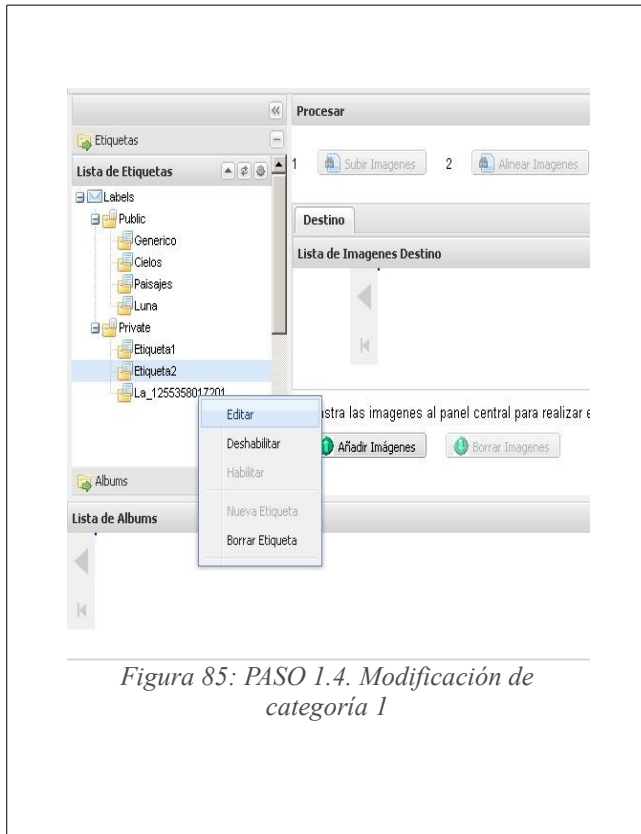


Figura 85: PASO 1.4. Modificación de categoría 1

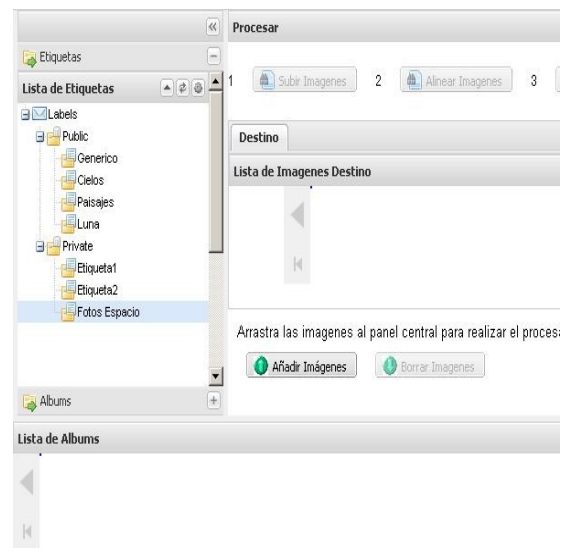


Figura 86: PASO 1.4. Modificación de categoría 2

El usuario debe seleccionar el nodo que desea modificar con el botón derecho del ratón y automáticamente le aparecerá un menú contextual en el que podrá seleccionar la opción de Editar la etiqueta.

Una vez ejecutada la operación el sistema permite que el usuario pueda introducir el nombre de la etiqueta que desea. En este caso se ha introducido el nombre “Fotos Espacio”.

PASO 1.5

El paso siguiente sería la asignación de la categoría al álbum previamente creado.

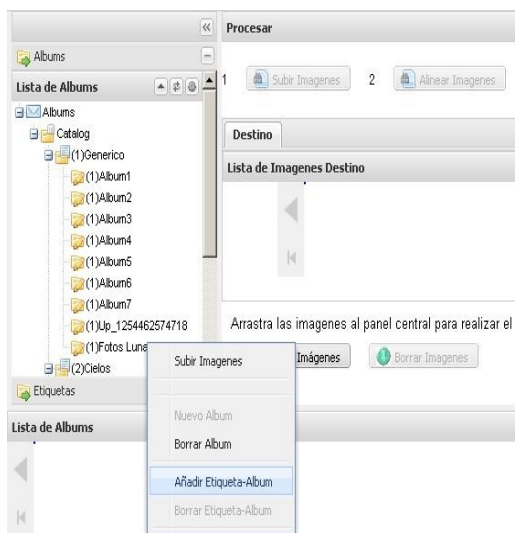


Figura 87: PASO5. Asignación de categoría 1

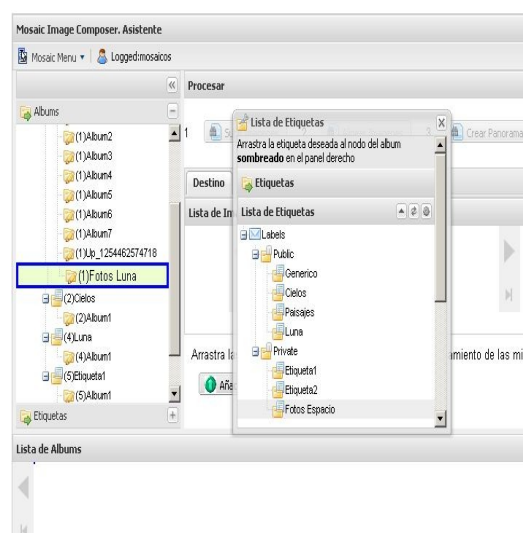


Figura 88: PASO5. Asignación de categoría 2

El usuario debe seleccionar el nodo que desea asignar con el botón derecho del ratón y automáticamente le aparecerá un menú contextual en el que podrá seleccionar la opción de Añadir categoría al álbum.

Una vez ejecutada la operación el álbum permanece sombreado para identificar claramente el elemento objetivo. Adicionalmente a la derecha de este panel se muestra un diálogo donde aparecen todas las etiquetas disponibles en el sistema entre las que está la que anteriormente se ha generado.

El usuario únicamente debe desplazar (drag-drop) la etiqueta deseada sobre el álbum sombreado. Automáticamente la categoría se añadirá al álbum y en el árbol aparecerá un nuevo nodo identificando la nueva categoría y el álbum.

En las siguientes ilustraciones se muestra el proceso completo.

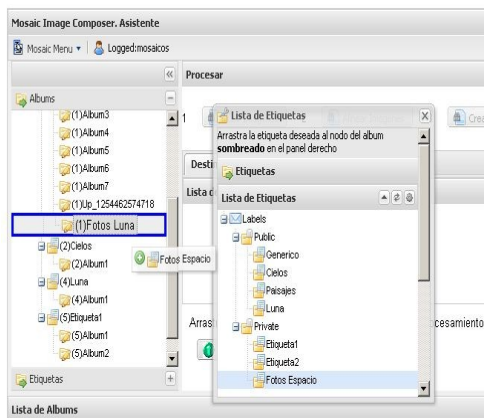


Figura 89: PASO 1.5. Asignación de categoría3

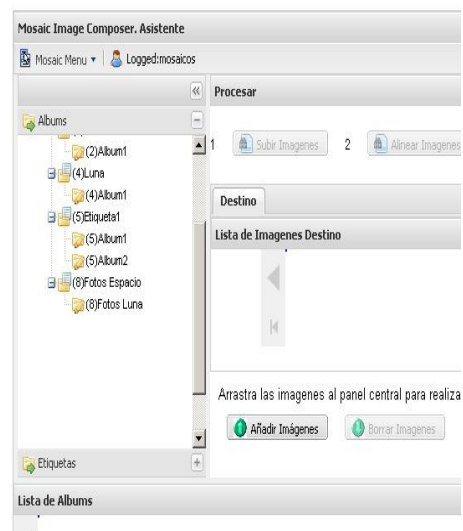


Figura 90: PASO 1.5. Asignación de categoría 4

6.3.2 PASO2. Carga de imágenes

Este paso se divide en dos subpasos que se explican a continuación.

PASO 2.1

Una vez que el álbum se ha creado y personalizado el siguiente paso consistirá en subir las imágenes deseadas al servidor para su proceso.

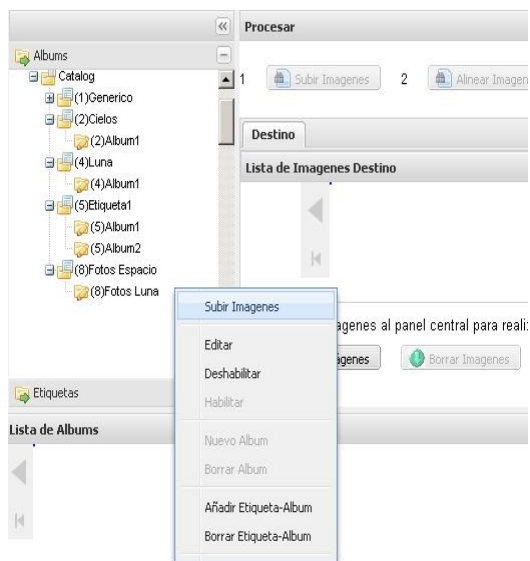


Figura 91: PASO 2.1. Carga de Imágenes 1

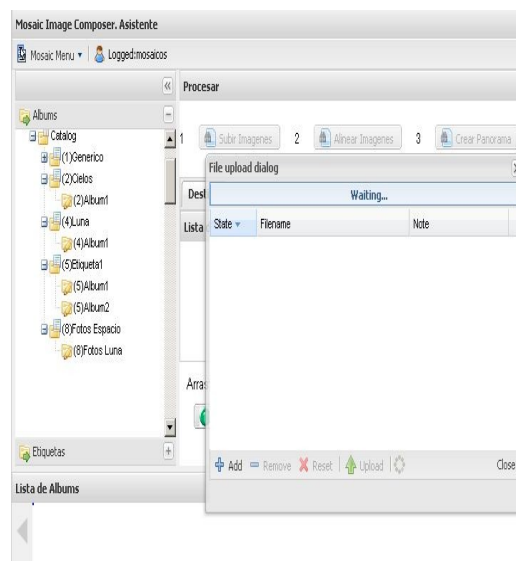


Figura 92: PASO 2.1. Carga de Imágenes 2

El usuario debe seleccionar en el álbum deseado (en nuestro caso **(8)Fotos_Luna**) con el botón derecho del ratón y automáticamente le aparecerá un menú contextual en el que podrá seleccionar la opción de Subir Imágenes.

Una vez ejecutada la operación a la derecha de este panel se muestra un diálogo donde el usuario puede incluir todas las imágenes que desea que disponga en su ordenador.

El usuario únicamente tendrá que pulsar el botón de Añadir Imágenes “Add” seleccionando de una en una las imágenes desde su ordenador.

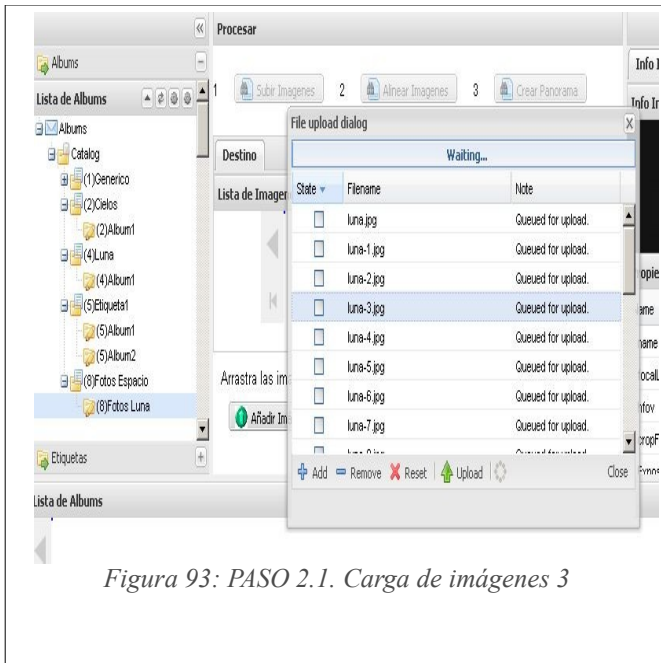


Figura 93: PASO 2.1. Carga de imágenes 3

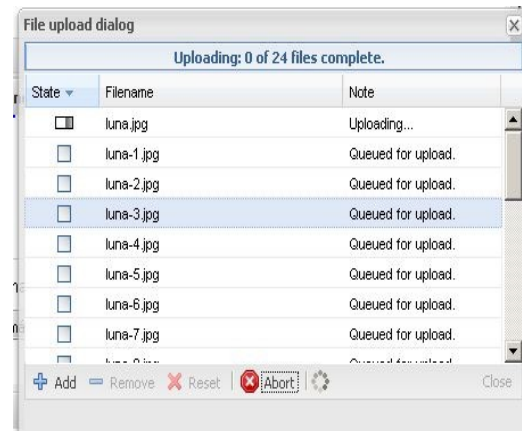


Figura 94: PASO 2.1. Carga de imágenes 4

Para cada imagen en el diálogo se mostrará el resultado de la subida de las mismas. En la siguiente ilustración se puede ver que todas las imágenes se han cargado correctamente.

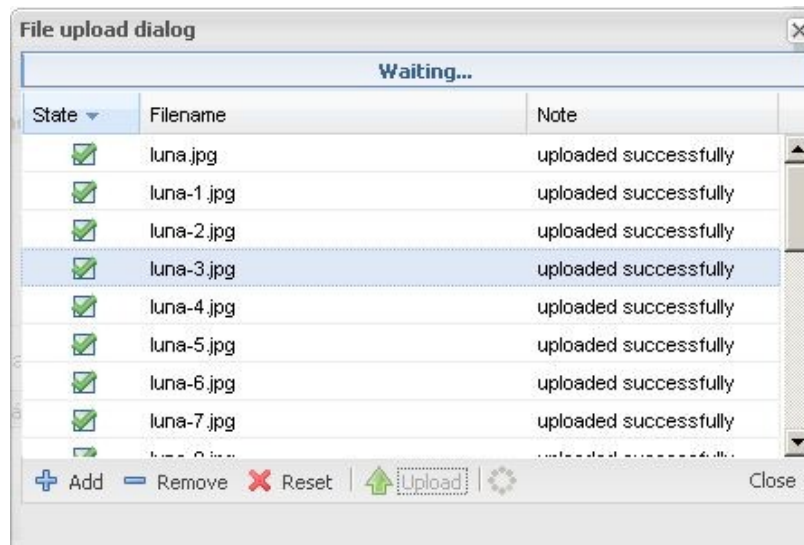


Figura 95: PASO 2.1. Carga de imágenes 5

PASO 2.2

Una vez subidas las imágenes al álbum a modo de comprobación el usuario puede seleccionar el álbum creado con el botón izquierdo del ratón y verificar si las imágenes se han cargado correctamente.



Figura 96: PASO 2.2. Verificación de Imágenes 1

Como se puede ver en la ilustración al pulsar el álbum creado en el panel inferior aparecen todas las imágenes previamente subidas al servidor.

6.3.3 PASO3. Subida de imágenes para composición

Una vez realizadas todas las operaciones el usuario ya puede realizar la composición del mosaico.

PASO 3.1

El primer paso que tiene que realizar es seleccionar las imágenes que desea que sean objetivo del proceso de alineación.

En nuestro caso las imágenes que se han introducido en el álbum “Fotos_Luna”. Aquí el usuario tiene dos posibilidades, arrastrar individualmente las imágenes que desea mediante la técnica de drag-drop del panel inferior al panel central o bien pulsar directamente el botón de “Añadir Imágenes” que añade todas y cada una de las imágenes del álbum.

La técnica de drag-drop permite incluso seleccionar varias imágenes a la vez mediante el uso de la tecla “Ctrl” del teclado y permite seleccionar únicamente las imágenes en las que estamos interesados.

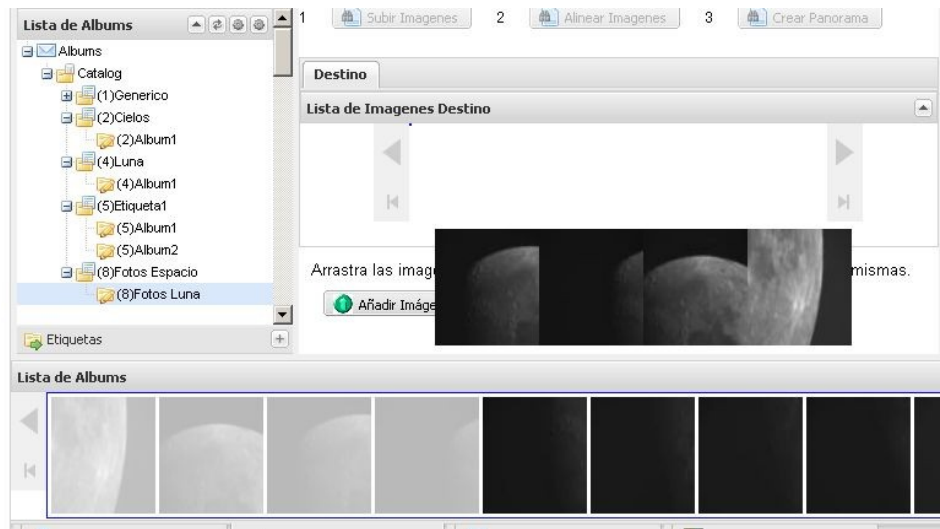


Figura 97: PASO 3.1 Subida de Imágenes para validación 1

Como resultado final de este paso el usuario podrá ver que las imágenes del panel inferior se han desplazado al panel central y las del panel inferior se han marcado como seleccionadas mediante un aspa roja. Si el usuario desea eliminar alguna imagen del panel central únicamente debería desplazarlas con el ratón al panel inferior y automáticamente ya no formaría parte del proceso de generación de mosaicos.

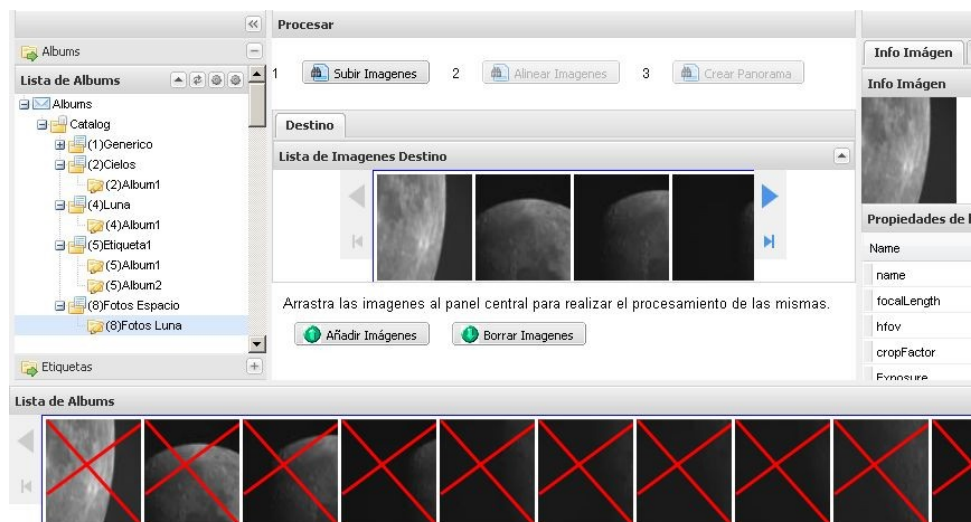


Figura 98: PASO 3.1 Subida de Imágenes para validación 2

PASO 3.2

El siguiente paso consiste en realizar una validación de las imágenes con el fin de verificar si son adecuadas para el proceso de generación de mosaicos. El sistema debe realizar una serie de verificaciones para comprobar si las imágenes tienen el formato adecuado y tienen información de metadatos para realizar el proceso de alineación.

Para ello el usuario deberá pulsar el botón de “**Subir Imágenes**” en el panel central. Como resultado de esta comprobación el sistema puede dar dos resultados, uno indicando que las imágenes son correctas y se puede continuar con el proceso de alineación y otro indicando que todas o alguna de las imágenes tienen un formato incorrecto.

Para este ejemplo el sistema detecta que las imágenes no tienen información de metadatos y que por tanto el usuario antes de continuar debe introducir dicha información. En la siguiente ilustración se muestra el resultado del proceso si se produce este error.

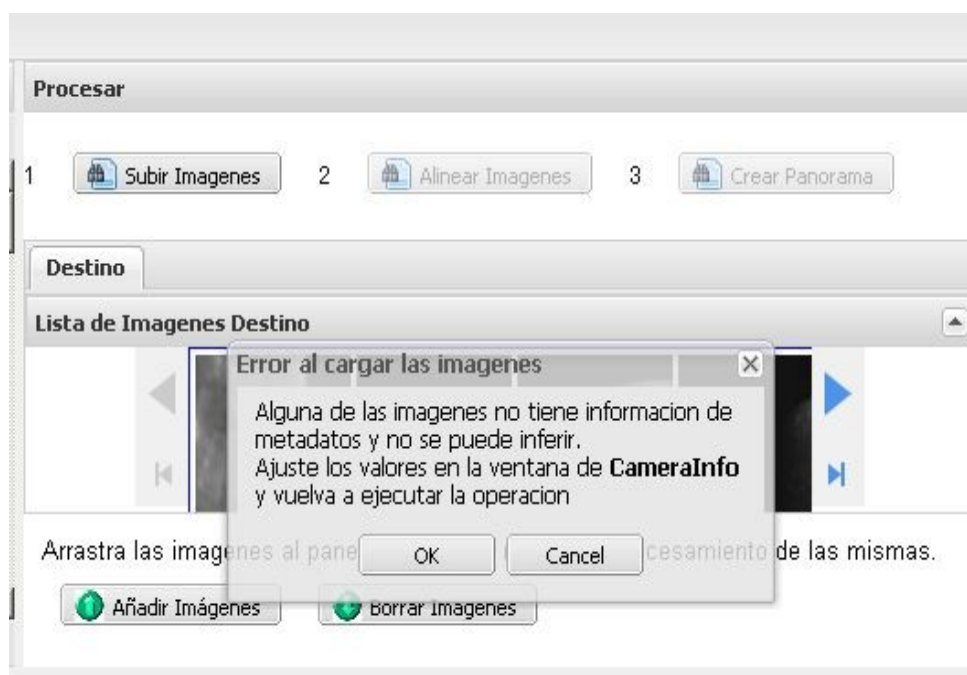


Figura 99: PASO 3.2 Verificación de imágenes 1

El botón del siguiente paso “Alinear Imágenes” y seguirá deshabilitado hasta que no se solucione el problema. El usuario por tanto debe seleccionar la pestaña “Información de Cámara” para introducir la información necesaria.

En esta pestaña el usuario puede ver que todas las imágenes que ha cargado carecen de los valores de tipo de lente, distancia focal y factor de recorte por lo que debe de consignarlos

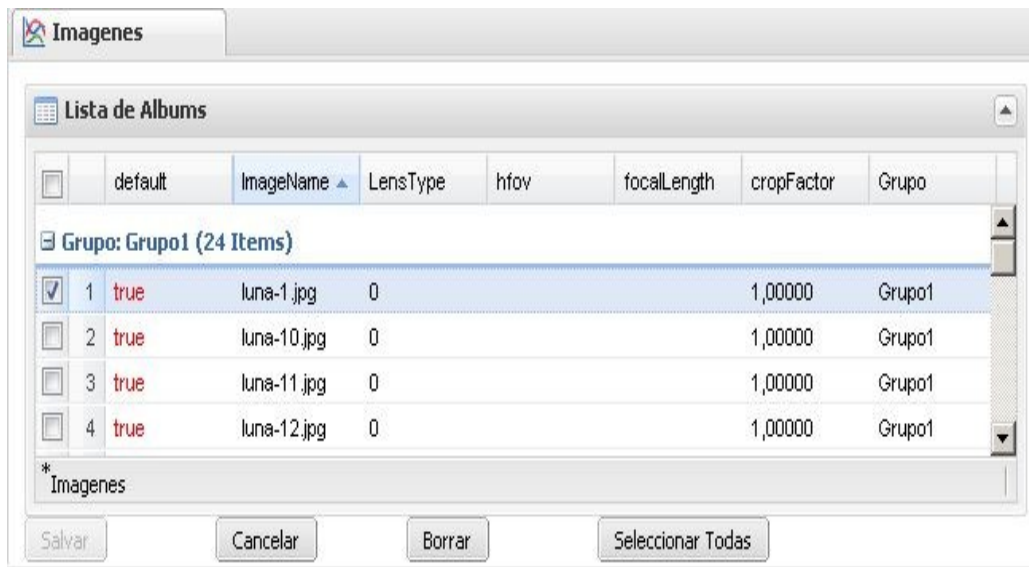


Figura 100: PASO 3.2 Verificación de imágenes 2

Para introducir esta información el usuario dispone de varias posibilidades. La primera consistiría en introducir los valores de forma individual para cada una de las imágenes, mientras que la segunda consistiría en aplicar una lente previamente almacenada y que ya dispone de los valores adecuados.

Para el primer caso el usuario si desea modificar todas las imágenes dispone de un botón “Seleccionar Todas” el cual marca todas las imágenes seleccionadas. Una vez realizada esta operación el usuario debe consignar los valores de distancia focal, hfov y cropFactor así como el tipo de lente con el que se ha tomado la imagen. Los valores de estos parámetros se pueden calcular automáticamente introduciendo simplemente alguno de ellos, por ejemplo introduciendo la distancia focal se calcula automáticamente el hfov y viceversa.

Para el segundo caso el proceso es más sencillo porque el usuario una vez seleccionada la imagen o grupo de imágenes a las que quiere aplicar la lente lo único que tiene que realizar es seleccionar dentro del desplegable de lentes la que más se adapta a sus características y pulsar el botón “Aplicar”. Considerando que muchas de la imágenes han sido tomadas por el telescopio del proyecto Ciclope Astro los valores de lente de ese telescopio ya se conocerán de antemano.

En las figura Figura 101 y Figura 102 se puede ver este proceso.

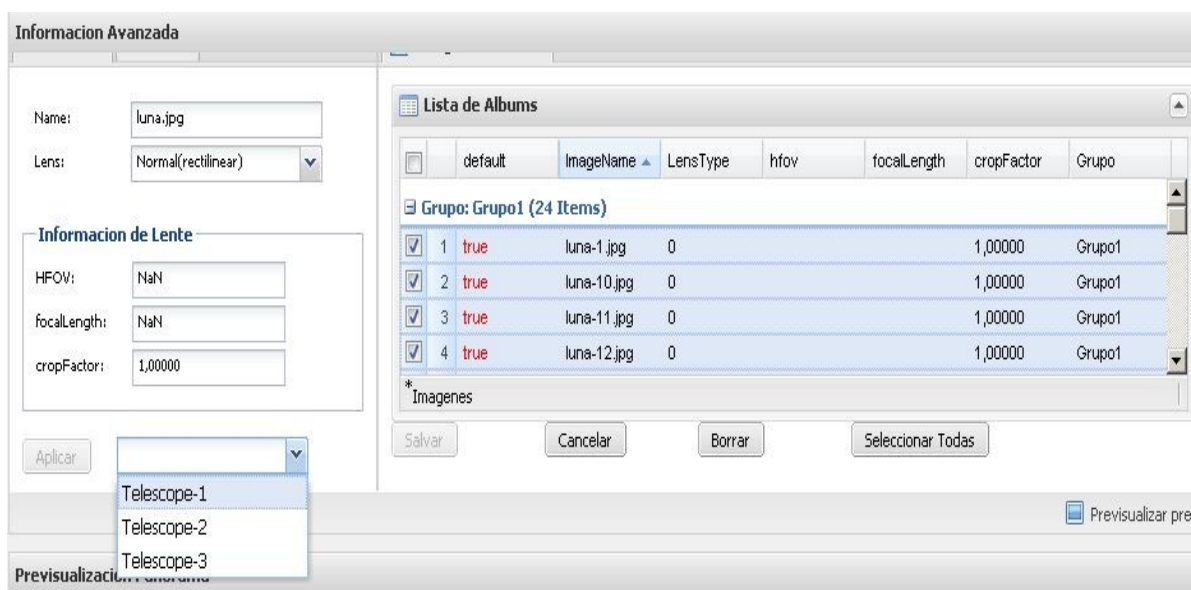


Figura 101: PASO 3.2 Verificación de imágenes 3

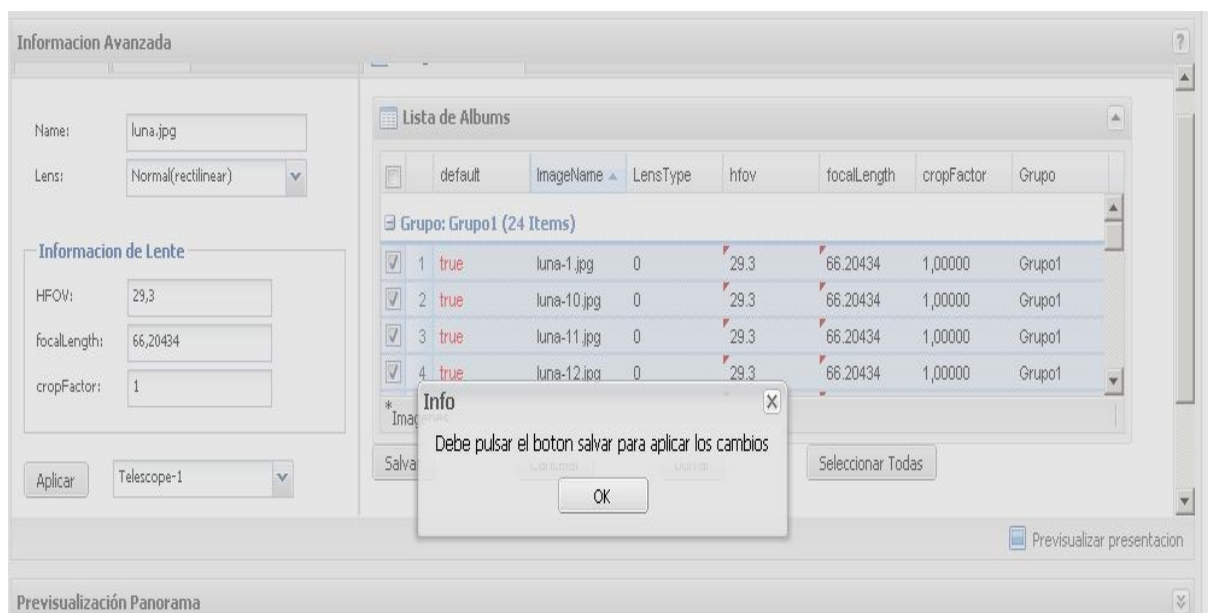


Figura 102: PASO 3.2 Verificación de imágenes 4

Una vez realizadas estas operaciones el usuario según le informa la aplicación debe aplicar los cambios y para ello deberá pulsar el botón “Guardar”. Si no se pulsa este botón las modificaciones aplicadas sobre las imágenes se perderán.

Realizadas estas operaciones el usuario debe volver a validar las imágenes para verificar que son correctas. Para ello deberá volver a la pestaña “Asistente” y volver a pulsar el botón “Subir Imágenes”.

El usuario puede ver en el panel derecho de información de imágenes la información individual de cada imagen. Situando el puntero del ratón sobre la imagen puede ver los valores aplicados de metadatos.

Si la validación se realiza correctamente el usuario recibirá una pantalla similar a la siguiente.

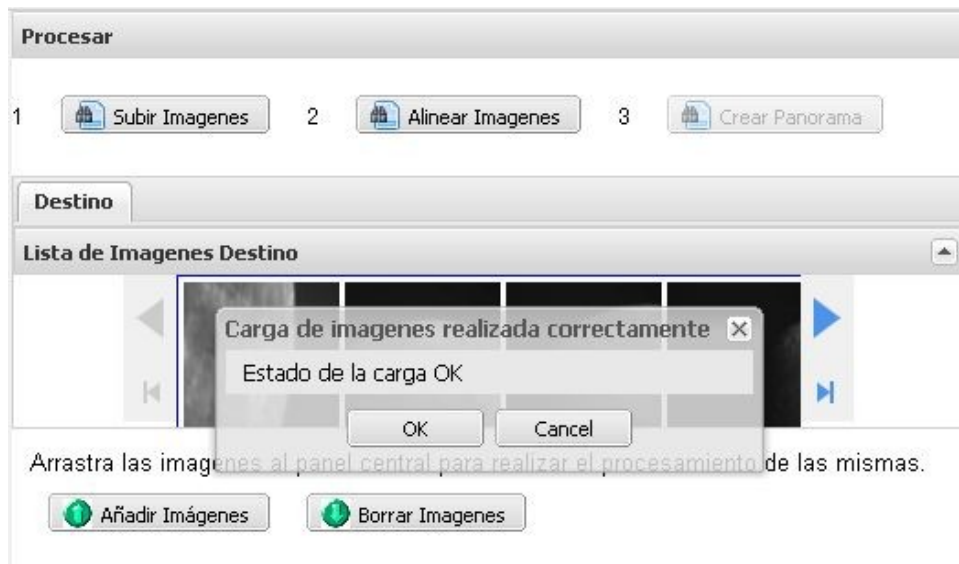


Figura 103: PASO 3.2 Verificación de imágenes 5

6.3.4 PASO4. Alineación de imágenes

Una vez que todas las imágenes están cargadas es preciso realizar el proceso de alineación. En este proceso las imágenes que se han subido al álbum y se han validado correctamente se analizan en búsqueda de características similares para realizar el proceso de composición.

PASO 4.1

Este proceso es el más costoso de todos porque el proceso de búsqueda de puntos invariantes comunes entre las imágenes utilizando algoritmos de reconocimiento de imágenes que necesita de una gran capacidad computacional. En este caso se utiliza el algoritmo SIFT que se ha comentado anteriormente en el trabajo.

El sistema una vez pulsado el botón de “Alinear Imágenes” muestra al usuario una consola donde se pueden ver todas las operaciones que se están realizando a nivel servidor para procesar las imágenes. Específicamente puede ir viendo el análisis SIFT que se realiza para cada foto con el fin de detectar los puntos comunes. Para cada imagen en la consola se puede ver el número de puntos significativos que encuentra con los cuales posteriormente va a realizar el proceso de alineación.

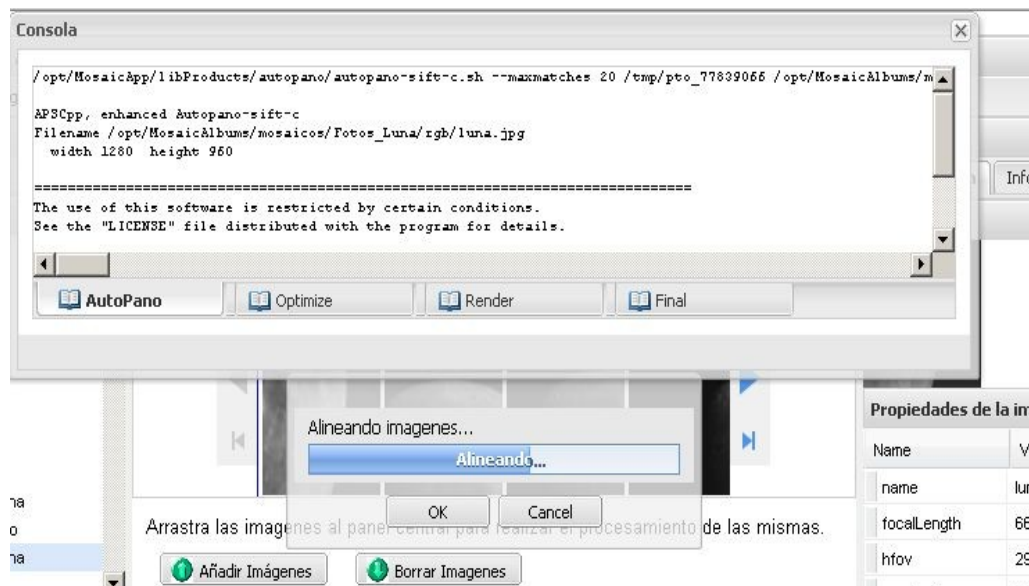


Figura 104: PASO 4.1. Alineación Imágenes 1

Una vez finalizado el análisis de las imágenes el siguiente paso interno consiste en verificar si existen puntos coincidentes entre las imágenes para poder generar el mosaico. Si existen puntos coincidentes el sistema genera la imagen y se puede previsualizar. En el ejemplo que estamos llevando a cabo existen una serie de imágenes que el sistema no puede conectar con el resto y es necesario realizar operaciones manuales.

PASO 4.2

En este caso existen una serie de imágenes que son demasiado oscuras y el algoritmo no puede detectar puntos comunes entre ellas. Concretamente las imágenes con identificadores ([1, 2, 3, 4, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]; [5]; [6]; [7]; [8]; [9]) El proceso de alineación no puede continuar hasta que se tome una decisión sobre esas imágenes

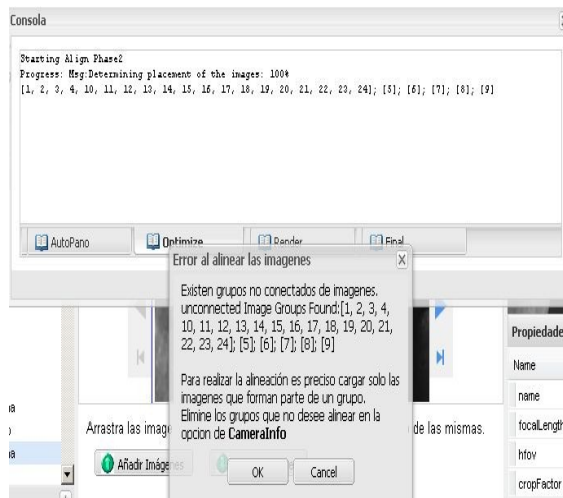


Figura 105: PASO 4.2. Arreglo Imágenes 2



Figura 106: PASO 4.2. Arreglo Imágenes 2

Lo más fácil en estos casos es eliminar aquellas imágenes que están provocando problemas para realizar una composición sin las mismas. Para ello el sistema sugiere al usuario que elimine los grupos de imágenes no conectados a través de la opción de “Información de Cámara”.

Para facilitar el proceso de borrado de imágenes el sistema agrupa las imágenes en el panel de información por grupos permitiendo saber cuantas imágenes componen cada grupo.

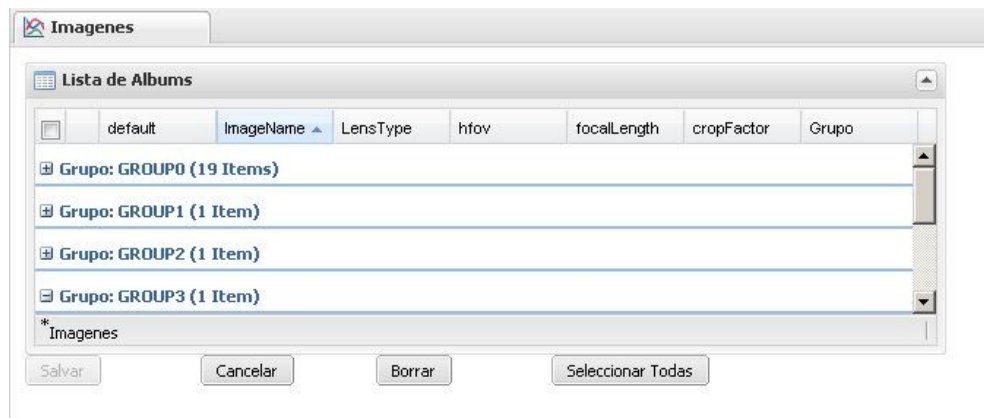


Figura 107: PASO 4.2 Arreglo Imagen 3

En la imagen se puede ver que existe un grupo “GROUP0” con 19 imágenes conectadas y luego existen grupos de 1 sola imagen. El usuario para poder continuar deberá borrar los grupos que tengan solo una imagen. Para ello simplemente deberá marcar el checkbox de la imagen de cada grupo como se puede ver en la siguiente figura y pulsar el botón “Borrar”. Como paso final una vez verificado que solo queda el grupo de 19 elementos deberá pulsar el botón “Salvar”.



Figura 108: PASO 4.2 Arreglo Imagen 4

Una vez realizada esta operación deberá volver a alinear las imágenes que han quedado conectadas. Este paso vuelve a realizar el proceso de búsqueda de puntos pero solo con las imágenes seleccionadas. El usuario puede ver en el panel central que las imágenes que se han borrado han desaparecido del panel de proceso y vuelven a estar para seleccionar en el panel inferior.

Si el proceso de alineación funciona correctamente el sistema deberá un mensaje indicando que se ha generado una imagen que el usuario puede previsualizar accediendo a la ventana de previsualización.

PASO 4.3

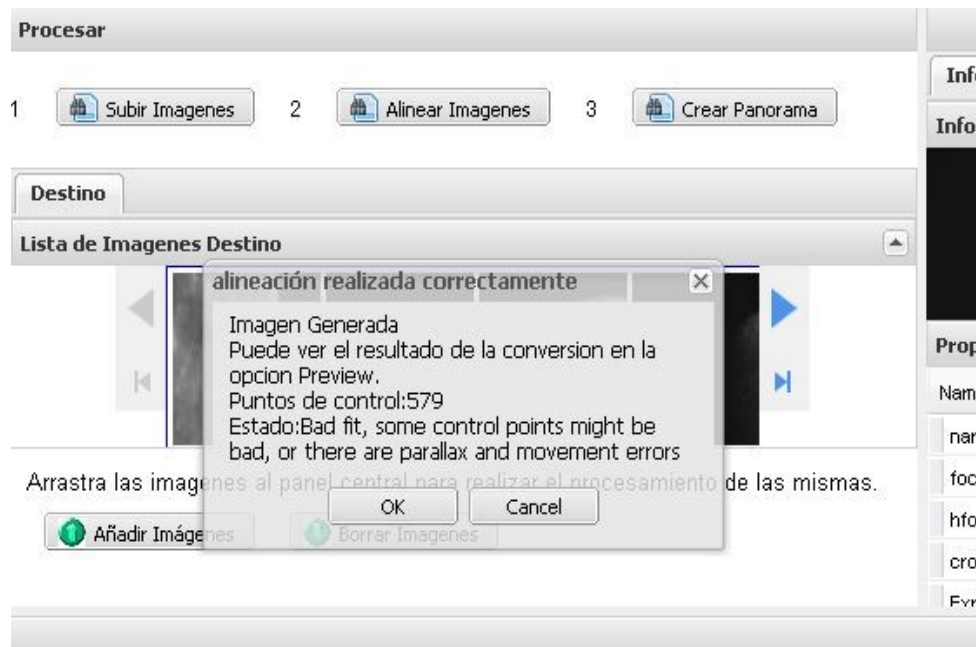


Figura 109: PASO 4.3 Previsualización Imagen 1

Como se puede ver en la siguiente ilustración el sistema indica que la imagen se ha generado correctamente, además de indicar el número de puntos de control con los que se ha generado la imagen y una indicación de cuan buena es la panorámica. En este caso nos indica que la panorámica no tiene una calidad demasiado alta.

El usuario una vez generada puede desplazarse a la ventana de **previsualización** de la panorámica para ver el resultado de la composición.

En la pantalla de información el usuario puede ver la URL de previsualización y de descarga por si quiere integrar esta información en otra aplicación y un botón para descargar la panorámica a disco.



Figura 110: PASO 4.3 Previsualización Imagen 2

Pulsando sobre la pestaña de “Previsualización del proceso de alineación” el usuario accede a una aplicación para realizar una previsualización de la información.

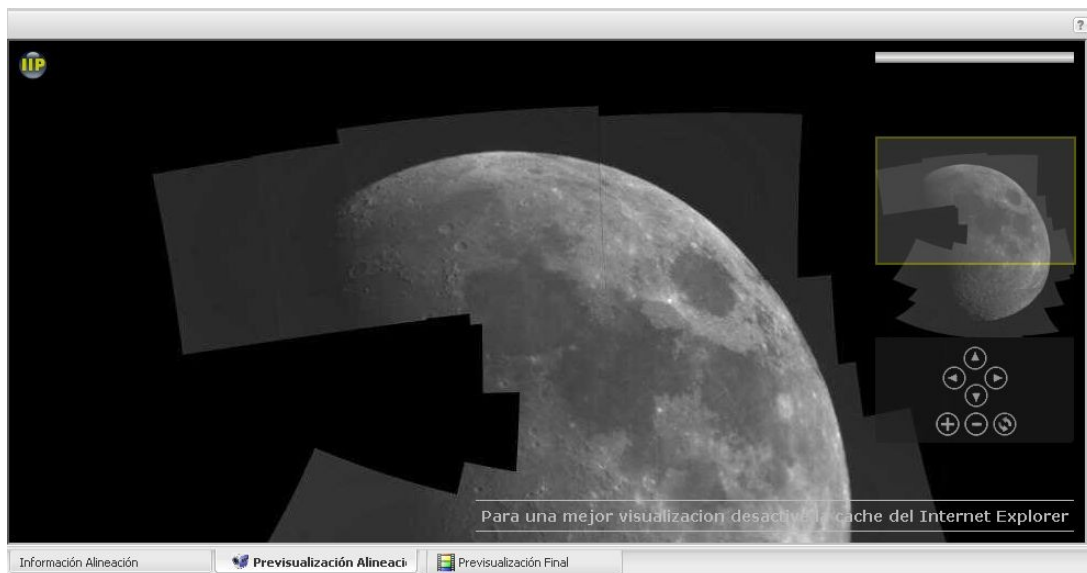


Figura 111: PASO 4.3 Previsualización Imagen 3

A través de esta opción el usuario puede ver la información de la imagen, desplazarse sobre la misma y realizar zoom utilizando una serie de controles en la zona derecha de la pantalla. También dispone de una imagen en pequeño en la zona derecha de la pantalla para desplazarse de forma más rápida a alguna zona específica de la imagen.

Como se puede ver en la imagen la zona izquierda en la que ve un hueco corresponde a las imágenes que se eliminaron de la panorámica por no tener la resolución adecuada.

6.3.5 PASO5. Generación de la panorámica

Una vez verificado que la imagen prevvisualizada es correcta el usuario deberá generar la imagen panorámica definitiva en la que se realiza una composición de una calidad más alta mediante la utilización de una serie de programas específicos de composición.

PASO 5.1

El usuario deberá pulsar el botón de “Generar Panorámica” identificado como el Paso 3 del proceso de composición y el sistema le mostrará de nuevo la consola en la que podrá ver las operaciones que se realizan para generar la panorámica.

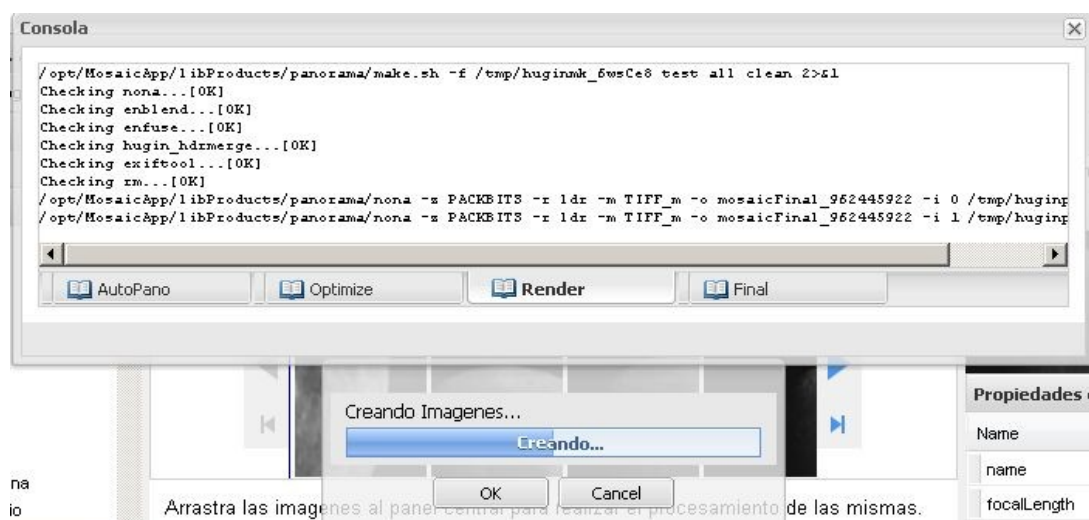


Figura 112: PASO 5.1. Generación Panorámica 1

En la consola el usuario puede ver los distintos filtros que se pasan para generar la panorámica y que se han comentado anteriormente.

- nona
- enblend
- enfuse
- hugin_hdrmerge

-
- exiftool

Si el resultado es correcto el usuario recibe una pantalla donde se indica el resultado final de la composición.

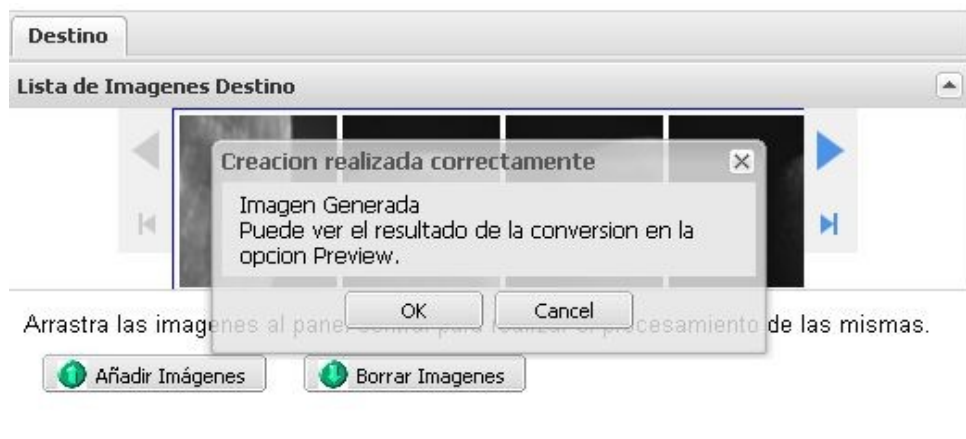


Figura 113: PASO 5.1. Generación Panorámica 2

PASO 5.2

El usuario una vez generada puede desplazarse a la ventana de **previsualización** de la panorámica igual que el paso anterior de previsualización para ver el resultado de la composición.

En la pantalla de información el usuario puede ver la URL final y de descarga por si quiere integrar esta información en otra aplicación y un botón para descargar la panorámica a disco. Como se puede ver en la ilustración el usuario sigue teniendo acceso a la información de previsualización.



Figura 114: PASO 5.2 Previsualización Imagen 1

Al igual que el caso anterior Pulsando sobre la pestaña de “Previsualización Final” el usuario accede a una aplicación para realizar una previsualización de la información.

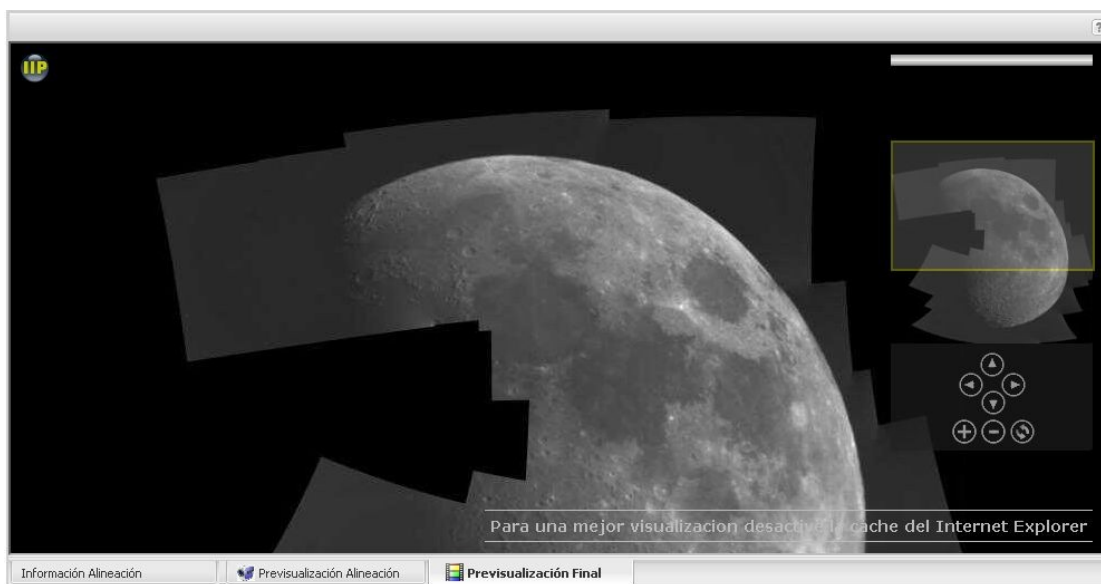


Figura 115: PASO 5.2 Previsualización Imagen 2

A través de esta opción el usuario puede ver la información de la imagen, desplazarse sobre la misma y realizar zoom utilizando una serie de controles en la zona derecha de la pantalla. También dispone de una imagen en pequeño en la zona derecha de la pantalla para desplazarse de forma más rápida a alguna zona específica de la imagen.



Figura 116: PASO 5.1. Previsualización Panorama 1

Como se puede ver en la imagen la zona izquierda en la que ve un hueco corresponde a las imágenes que se eliminaron de la panorámica por no tener la resolución adecuada.

6.3.6 PASO6. Resultado Final

La panorámica en este paso ya está terminada y está asociada al álbum de forma definitiva permitiendo que en todo momento el usuario pueda acceder a la información de la panorámica generada.

Pulsando sobre el álbum en cualquier momento el usuario puede ver en el panel derecho la panorámica que se generó la última vez y puede descargársela.



Figura 117: PASO 6.1. Resultado Final 1

7 Manual de instalación

En este punto se explica la instalación de todas las herramientas necesarias para poder ejecutar el sistema de generación de mosaicos. Se centrará esta implantación en el sistema operativo Linux concretamente en la distribución Ubuntu 9.0.4

Se procederá a explicar en detalle la instalación de todos los paquetes de software necesarios para poder ejecutar la herramienta y que tendremos que descargar previamente de sus sitios web correspondientes.

El trabajo realizado en este proyecto se ha llevado a cabo utilizando una tecnología que está teniendo un gran auge en los últimos años, nos estamos refiriendo a la tecnología de virtualización. La virtualización permite transformar “hardware” en “software”. Mediante software específico se transforman o “virtualiza” los recursos de hardware de un ordenador, incluidos CPU, RAM, disco duro y controlador de red, para crear una máquina virtual completamente **funcional** que puede ejecutar su propio sistema operativo y aplicaciones de la misma forma que lo hace un ordenador “real”.

Varias máquinas virtuales comparten recursos de hardware sin **interferir** entre sí de modo que se pueden ejecutar simultáneamente y de forma segura varios sistemas operativos y aplicaciones en un único ordenador completamente diferente, denominado máquina virtual.

Existen un buen número de programas que nos van a permitir crear máquinas virtuales, a continuación se muestran los más populares:

- **VMWare:** Es el estándar del mercado. Actualmente ofrece una versión gratuita.
- **QEMU:** Software de virtualización gratuito. Muy popular en el mundo Linux.
- **Virtual PC:** La alternativa de Microsoft. Se incluirá en las versiones Server de Windows Vista. También es gratuito.
- **BOCHS:** Software de virtualización gratuito. Pretende ser la alternativa del software libre a VMWare
- **Virtual Box:** El producto de Sun Microsystems. Además es Open Source.

La elección del software en este caso se ha basado básicamente en la experiencia y el conocimiento del alumno en la plataforma VMWare más que en las características de los productos. Concretamente se ha utilizado el producto **VMWare Server** versión gratuita de la Suite VMWare.

De esta forma se han generado varios entornos de desarrollo que han permitido poder realizar pruebas de la plataforma sin tener que disponer de un entorno Linux dedicado para realizar las pruebas.

Concretamente se han generado **dos entornos independientes**, uno para la compilación de las librerías y herramientas C++ y otro para la ejecución del software.

La ventaja de esta separación radica en que para el entorno de compilación es necesario descargar

muchas librerías con unas versiones determinadas y por tanto necesita de una instalación casi dedicada a esa compilación. En el entorno de instalación únicamente se distribuyen las librerías dinámicas necesarias así como los productos necesarios sobre los que instalar el software.

Para ambos entornos se ha utilizado la distribución Ubuntu 9.0.4 con una instalación básica instalada con soporte de compilación. La descarga se ha realizado desde la Web de Ubuntu (<http://www.ubuntu.com/getubuntu>). La instalación del sistema operativo queda fuera del alcance de este documento.

Para ambos sistemas se ha creado un usuario de sistema denominado **mosaicos** perteneciente al grupo **mosaicos** con el que se van a realizar todas las operaciones para evitar en la medida de lo posible la utilización del usuario **root**.

7.1 Manual de compilación

El aplicativo disponible en esta versión de la aplicación de mosaicos está compuesto como se ha ido comentado a lo largo del documento por una serie de módulos desarrollados tanto en la parte cliente como servidora de la aplicación. Concretamente se han desarrollado los siguientes componentes cuyo proceso de compilación se comenta en los siguientes apartados.

1. Interfaz de cliente desarrollado con la tecnología GWT en lenguaje de programación Java.
2. Módulo servidor para la gestión de los álbumes y las imágenes e interfaz contra el sistema de composición de mosaicos.
3. Librería dinámica para la generación de mosaicos basada en el software de composición hugin y sus dependencias entre las que están las panorama tools.

Para realizar la compilación del aplicativo es necesario además disponer de un conjunto de librerías adicionales que se descargan como paquetes adicionales de la distribución Ubuntu.

7.1.1 Requisitos

Prevía a la preparación del entorno es preciso instalar los siguientes componentes:

El JDK de Sun permite la ejecución de aplicaciones Java así como la compilación de código. Se necesita descargar por tanto una versión completa del JDK de Sun.

Para instalarlo hay que descargar antes que nada el código del JDK que se encuentra disponible en <http://java.sun.com/javase/downloads/index.jsp>. En este caso descargaremos el binario de la versión de Linux 32 bits.

La versión descargada era la más estable en el momento de preparación del entorno de instalación.

Versión	Fichero
1.5.0_17	jdk1.5.0_17.tgz

El software descargado lo situaremos en el siguiente directorio previa a la descompresión del mismo **1.JDK**.

```
/opt/MosaicApp/MosaicSoftware.src/1.JDK
```

Una vez descargado lo descomprimiremos utilizando los siguientes comandos:

```
shell>cd /opt/MosaicApp
```

```
shell>gzip -cd /opt/MosaicApp/MosaicSoftware.src/1.JDK/jdk1.5.0_17.tgz |tar xvf -
```

Esto generará un directorio denominado /opt/MosaicApp/jdk1.5.0_17 localización final de la máquina virtual.

7.1.2 Compilación librerías mosaicos

El producto desarrollado para la generación de mosaicos en el servidor es una librería dinámica **“libMosaic.so”** generada a partir de diferentes módulos desarrollados y librerías externas. Esta librería ofrece una interfaz JNI (Java Native Interface) accesible desde una aplicación Java.

Librería	Versión	Fichero
libMosaic	1.1	libMosaic.src-1.1.tar.gz

Para la generación de esta librería es preciso realizar la compilación adicional del siguiente software base cuyo proceso de instalación se describe a continuación..

Librería	Versión	Fichero
Hugin	V3708	Software descargable por SVN + más dependencias hugin-patched.src-v3708.tar.gz
Boost	1.39	boost_1_39_0.tgz
Process	0.31	process_0.31.tgz

Estas librerías que se proporcionan como parte del proyecto se distribuirán en la página Web del proyecto SourceForge para que pueda ser descargado por cualquier usuario.

En los siguiente pasos se comentará el proceso de compilación de esta distribución adaptada (hugin) así como de la librería generada (libMosaic.so). En primer lugar se comenzará con las dependencias necesarias para el software.

Existe un tutorial muy interesante donde indica los pasos necesarios para realizar una compilación del producto directamente del CVS en la siguiente URL (http://wiki.panotools.org/Hugin_Compiling_ubuntu).

En los siguientes apartados se comentarán en orden la compilación e instalación de los siguientes productos.

- Hugin y sus dependencias.
- Librerías Boost.

-
- Librería libMosaic (La desarrollada en este proyecto).

7.1.2.1 Hugin

Previo a la compilación del producto es necesario realizar la instalación de un conjunto de librerías si no están disponibles en el equipo.

- Enblend
- libpano13
- autotools
- autopano-sift-c
- exiftool

7.1.2.1.1 Entorno inicial

A parte de estas librerías es preciso instalar un conjunto de paquetes de apoyo para permitir la compilación de estas herramientas (compiladores, control de versiones, librerías gráficas, etc.) que se distribuyen como paquetes de Ubuntu.

- Paquetes para poder compilar y generar las librerías dinámicas

```
shell>sudo apt-get install build-essential autoconf automake1.9 libtool flex bison gdb
```

- Libc6

```
shell>sudo apt-get install libc6-dev libgcc1
```

- Software de control de versiones svn

```
shell>sudo apt-get install subversion mercurial subversion-tools
```

7.1.2.1.2 Enblend

Enblend es una herramienta de composición de imágenes. Dado un conjunto de imágenes que se superponen de alguna forma irregular, Enblend las superpone de tal manera que la unión entre las imágenes es invisible, o al menos muy difícil de ver.

- Librerías gráficas para manejo de imágenes.

```
shell>sudo apt-get install pkg-config libtiff4-dev libboost-graph-dev libboost-thread-dev  
liblcms1-dev libglew1.5-dev libplot-dev libglut3-dev libopenexr-dev libxi-dev libxmu-dev
```

```
help2man texi2html texinfo
```

- Una vez descargas las dependencias nos descargamos el software de enblend

```
shell>mkdir -p /opt/MosaicApp/MosaicImageComposer/dependencies

shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies

shell>hg clone http://enblend.hg.sourceforge.net:8000/hgroot/enblend/enblend enblend
```

- y lo compilamos e instalamos.

```
shell>cd enblend

shell>make --makefile=Makefile.scm

shell>mkdir BUILD

shell>cd BUILD

shell>CXXFLAGS="--param inline-unit-growth=60 -march=native -O2" ../configure --disable-
image-cache --enable-openmp

shell>make

shell>sudo make install
```

7.1.2.1.3 libpano13

Libpano13 son las librerías que implementan las panotools.

- Librerías gráficas para manejo de imágenes.

```
shell>sudo apt-get install zlib1g zlib1g-dev libpng12-dev libjpeg62-dev libtiff4-dev
```

- Una vez descargas las dependencias nos descargamos el software.

```
shell>mkdir -p /opt/MosaicApp/MosaicImageComposer/dependencies

shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies

shell>svn co https://panotools.svn.sourceforge.net/svnroot/panotools/trunk/libpano libpano13
```

- y lo compilamos

```
shell>mkdir build.libpano

shell>cd build.libpano

shell>cmake ../libpano13 -DCMAKE_INSTALL_PREFIX=/usr/local -DCPACK_BINARY_DEB:BOOL=ON
-DCPACK_BINARY_NSIS:BOOL=OFF -DCPACK_BINARY_RPM:BOOL=OFF -DCPACK_BINARY_STGZ:BOOL=OFF
-DCPACK_BINARY_TBZ2:BOOL=OFF -DCPACK_BINARY_TGZ:BOOL=OFF -DCPACK_BINARY_TZ:BOOL=OFF
```

```
shell>make package

shell>sudo dpkg -i libpano13-*--Linux.deb
```

7.1.2.1.4 autopano-sift-c

Autopano es el software que permite obtener los puntos dentro de las imágenes que permiten detectar características invariantes en las imágenes. Para utilizar debemos descargarnos previamente un conjunto de paquetes para su compilación.

- Librerías gráficas.

```
shell>sudo apt-get install libxml2-dev
```

- Una vez descargas las dependencias nos descargamos el software.

```
shell>mkdir -p /opt/MosaicApp/MosaicImageComposer/dependencies

shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies

shell>svn co https://hugin.svn.sourceforge.net/svnroot/hugin/autopano-sift-C/trunk/
autopano-sift-C
```

- y lo compilamos

```
shell>mkdir autopano-sift-C.build

shell>cd autopano-sift-C.build

shell>cmake .. /autopano-sift-C -DCMAKE_INSTALL_PREFIX=/usr/local
-DPACK_BINARY_DEB:BOOL=ON -DPACK_BINARY_NSIS:BOOL=OFF -DPACK_BINARY_RPM:BOOL=OFF
-DPACK_BINARY_STGZ:BOOL=OFF -DPACK_BINARY_TBZ2:BOOL=OFF -DPACK_BINARY_TGZ:BOOL=OFF
-DPACK_BINARY_TZ:BOOL=OFF

shell>make package

shell>sudo dpkg -i autopano-sift-C-*--Linux.deb
```

7.1.2.1.5 exiftool

Exiftool es una herramienta para leer y escribir información de metadatos en las imágenes.

- Desinstalamos cualquier versión anteriormente disponible.

```
shell>sudo apt-get purge libimage-exiftool-perl
```

- Una vez descargas las dependencias nos descargamos el software.

```
shell>mkdir -p /opt/MosaicApp/MosaicImageComposer/dependencies

shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies
```

```
shell>wget http://www.sno.phy.queensu.ca/~phil/exiftool/Image-ExifTool-7.94.tar.gz
```

- y lo compilamos

```
shell>tar -xzf Image-ExifTool-7.94.tar.gz

shell>cd Image-ExifTool-7.94

shell>perl Makefile.PL

shell>make test

shell>sudo make install
```

7.1.2.1.6 Hugin

Para la compilación de Hugin es necesario un conjunto de librerías adicionales.

- Librerías gráficas para manejo de imágenes.

```
shell>sudo apt-get install cmake libopenexr-dev libboost-dev boost-build libboost-thread-dev
libboost-graph-dev gettext libwxgtk2.8-dev libxiv2-dev libimage-exiftool-perl libglew-dev
liblapack-dev
```

El software descargado (**hugin-adapted.src-v3708.tar.gz**) lo situaremos en el siguiente directorio previa a la descompresión del mismo **8.Hugin**.

El software **hugin (hugin-patched.src-v3708.tar.gz)** contiene las modificaciones realizadas para este proyecto (concretamente la solución a un problema de memoria)

- Stitcher.h
 - src/hugin_base/nona/Stitcher.h

En este proyecto el software de hugin se descarga vía svn ya que se necesita una versión específica (v3708). Se puede ver la versión mediante la ejecución de este comando. Una vez descargada la versión es preciso aplicar el parche para solucionar el problema.

```
shell> svn info

Ruta: .

URL: https://hugin.svn.sourceforge.net/svnroot/hugin/hugin/trunk

Raíz del repositorio: https://hugin.svn.sourceforge.net/svnroot/hugin

UUID del repositorio: 2038758d-bd2d-0410-8553-d98df698c6e5

Revisión: 3708
```

```
Tipo de nodo: directorio
Agendado: normal
Autor del Ãºltimo cambio: stativ
RevisiÃ³n del Ãºltimo cambio: 3708
Fecha de Ãºltimo cambio: 2009-03-06 14:41:42 +0100 (vie 06 de mar de 2009)
```

Si se desea la última versión disponibles en el repositorio y trabajar con ella el comando que habría que ejecutar sería el siguiente:

```
svn co https://hugin.svn.sourceforge.net/svnroot/hugin/hugin/trunk hugin
```

Si se desea descargar directamente del svn la versión con la que se ha trabajado es necesario ejecutar este comando.

```
svn co -r 3708 https://hugin.svn.sourceforge.net/svnroot/hugin/hugin/trunk hugin.3708
```

Una vez copiado el parche nos descargaremos del svn la revisión de trabajo y le aplicamos el parche.

```
shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies
shell>svn co -r 3708 https://hugin.svn.sourceforge.net/svnroot/hugin/hugin/trunk hugin
shell>gzip -cd /opt/MosaicApp/MosaicSoftware.src/8.Hugin/hugin-adapted.src-v3708.tar.gz |tar
xvf -
```

Esto generará un directorio denominado /opt/MosaicApp/MosaicImageComposer/dependencies/hugin localización del producto para su compilación.

Los siguientes pasos permiten compilar el producto.

```
shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies
shell>mkdir hugin-build
shell>cd hugin-build
shell>cmake ../hugin -DENABLE_LAPACK=YES -DCPACK_BINARY_DEB:BOOL=ON
-DCPACK_BINARY_NSIS:BOOL=OFF -DCPACK_BINARY_RPM:BOOL=OFF -DCPACK_BINARY_STGZ:BOOL=OFF
-DCPACK_BINARY_TBZ2:BOOL=OFF -DCPACK_BINARY_TGZ:BOOL=OFF -DCPACK_BINARY_TZ:BOOL=OFF
shell>make package
shell>sudo dpkg -i hugin-*-Linux.deb
```

7.1.2.1.7 PanoGLView

PanoGwView es una herramienta para acelerar el procesamiento de imágenes.

-
- Instalamos los requisitos.

```
shell>sudo apt-get install wx-common
```

- Una vez descargas las dependencias nos descargamos el software.

```
shell>mkdir -p /opt/MosaicApp/MosaicImageComposer/dependencies  
  
shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies  
  
shell>svn co https://hugin.svn.sourceforge.net/svnroot/hugin/panoglview/trunk panoglview
```

- y lo compilamos.

```
shell>cd panoglview  
  
shell>./bootstrap  
  
shell>./configure  
  
shell>make  
  
shell>sudo make install
```

7.1.2.2 Boost-Process

El sistema utiliza una serie de librerías para la gestión de procesos (<http://www.boost.org/>). Es un conjunto de librerías multiplataforma que permite disponer de un conjunto de algoritmos, funciones matemáticas, procesamiento de entrada y salida de datos para extender las librerías C++.

Las librerías son las siguientes.

Librería	Versión	Fichero
Boost	1.39	boost_1_39_0.tgz
Process	0.31	process_0.31.tgz

- Las librerías se pueden descargar de la web del autor pero se proporcionan como parte del proyecto ya que concretamente en la librería process se ha realizado una adaptación. El software descargado (**boost_1_39_0.tgz** y **process_0.31.tgz**) lo situaremos en el siguiente directorio previa a la descompresión del mismo **9.libMosaic**.

```
shell>mkdir -p /opt/MosaicApp/MosaicImageComposer/dependencies  
  
shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies  
  
shell>tar xzvf /opt/MosaicApp/MosaicSoftware.src/9.libMosaic/boost_1_39_0.tgz  
  
shell>tar xzvf /opt/MosaicApp/MosaicSoftware.src/9.libMosaic/process_0.31.tgz
```

- Instalamos librerías externas que no se proporcionan en la distribución.

```
shell>sudo aptitude install libboost-system1.37.0

shell>sudo ln -s /usr/lib/libboost_system-mt.so.1.37.0 /usr/lib/libboost_system-mt.soN
```

- Para este caso no es necesario realizar ninguna compilación adicional. Esto generará dos directorios que deberemos luego identificar en el proceso de compilación de la librería libMosaic.so
 - /opt/MosaicApp/MosaicImageComposer/dependencies/boost_1_39_0
 - /opt/MosaicApp/MosaicImageComposer/dependencies/process_0.31

7.1.2.3 LibMosaic

Como último paso para montar el entorno de compilación es preciso disponer de la librería desarrollada como parte del proyecto. El resto de paquetes comentados hasta el momento son paquetes adicionales necesarias para poder compilar este producto. Básicamente estas librería es un recubrimiento JNI de una serie de funciones del hugin y de las librería panotools que permiten que pueda ser llamada desde una aplicación Java.

Las librerías son las siguientes.

Librería	Versión	Fichero
libMosaic	1.1	libMosaic.src-1.1.tar.gz

- El software descargado (**libMosaic.src-1.1.tar.gz**) lo situaremos en el siguiente directorio previa a la descompresión del mismo **9.libMosaic**.

```
shell>mkdir -p /opt/MosaicApp/MosaicImageComposer/dependencies

shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies

shell>tar xzvf /opt/MosaicApp/MosaicSoftware.src/9.libMosaic/libMosaic.src-1.1.tar.gz
```

- Una vez descomprimido se generará el siguiente directorio previo a la compilación.

```
/opt/MosaicApp/MosaicImageComposer/libMosaic.src
```

- Si se desea en lugar de descomprimir el fichero, el usuario también puede descargarse el software del repositorio SVN.

```
shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies

shell>cd /opt/MosaicApp/MosaicImageComposer/dependencies

shell>svn
```

co

```
http://tornasol.datsi.fi.upm.es/ciclope/ciclopesvn/ciclope_contribution/tfc/2009fjgarcia/Software/Mosaicos/Modulos/ModulosC++/MosaicApp/src/libMosaic.src
```

- El siguiente paso será la compilación de la aplicación. En este directorio generamos creamos un enlace simbólico denominado current hacia el directorio v1.1. En primer lugar deberemos verificar si los directorios de dependencias de la aplicación están correctamente identificados. Estas dependencias se almacenan en el fichero

```
vi /opt/MosaicApp/MosaicImageComposer/libMosaic.src/current/CMakeFiles/mosaic.dir/flags.make

#####

JNI_INCLUDE_FLAGS=-I/opt/MosaicApp/jdk1.5.0_17/include
-I/opt/MosaicApp/jdk1.5.0_17/include/linux

BOOST_FLAGS=-I/opt/MosaicApp/MosaicImageComposer/dependencies/process_0.31
-I/opt/MosaicApp/MosaicImageComposer/dependencies/boost_1_39_0

HUGIN_HOME=/opt/MosaicApp/MosaicImageComposer/dependencies/hugin

DEPEND=/opt/MosaicApp/MosaicImageComposer/dependencies

HUGIN_HOME_BUILD=/opt/MosaicApp/MosaicImageComposer/dependencies/hugin-build

MOSAIC_HOME=-I/opt/MosaicApp/MosaicImageComposer/libMosaic.src/current

CXX_FLAGS = ${JNI_INCLUDE_FLAGS} ${BOOST_FLAGS} ${MOSAIC_HOME} -I/usr/lib/wx/include/gtk2-
unicode-release-2.8 -I/usr/include/wx-2.8 -pthread -O2 -g -I${HUGIN_HOME}/src -I$
{HUGIN_HOME}/src/hugin_base -I${HUGIN_HOME}/src/foreign -I${HUGIN_HOME}/src/foreign/vigra -I$
{HUGIN_HOME}/src/celeste -I/usr/include/OpenEXR -I/usr/local/include -isystem $
/usr/lib/wx/include/gtk2-unicode-release-2.8 -isystem /usr/include/wx-2.8 -isystem $
{HUGIN_HOME}/src/-pthread -I${HUGIN_HOME}/src/hugin1 -I${HUGIN_HOME_BUILD}/src
-D_FILE_OFFSET_BITS=64 -D_LARGE_FILES -D_WXGTK__ -pthread
```

- Seguidamente compilamos la aplicación

```
shell>cd /opt/MosaicApp/MosaicImageComposer/libMosaic.src/

shell>ln s v1.1 current

shell>cd /opt/MosaicApp/MosaicImageComposer/libMosaic.src/current

shell>make clean

shell>make

shell>make install
```

- El resultado es una librería dinámica y un conjunto de librerías de productos que deberemos distribuir para que posteriormente pueda ser instalada.

```
Shell> make distrib

(EL SCRIPT BASICAMENTE HACE LO SIGUIENTE)

shell>cd /opt/MosaicApp/MosaicImageComposer
```

```
shell>tar czvf libMosaic-1.1.tar.gz libMosaic

shell>mkdir -p /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer

shell>cp libMosaic-1.1.tar.gz /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer

shell>tar czvf libMosaicProducts-1.1.tar.gz libMosaicProducts

shell>cp                                     libMosaicProducts-1.1.tar.gz
/opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer
```

7.1.3 Compilación módulo cliente/servidor GWT

El módulo cliente desarrollado con tecnología GWT en lenguaje de programación Java está disponible en el repositorio de SourceForge y puede ser descargado para su compilación. Este módulo necesita de unas librerías también desarrolladas como parte del proyecto y que necesitan ser compiladas para generar los ficheros JAR.

7.1.3.1 Requisitos

Como requisitos para la compilación del proyecto es necesario descargarse los siguientes paquetes. Si la instalación que se descarga mediante paquetes es inferior a la versión 2.2.1 (mvn -versión) habrá que instalar una versión directamente mediante paquetes descargados de la web de maven.

Paquete	Comando de búsqueda/instalación
Maven2 (Version 2.2.1)	sudo aptitude search maven2 sudo aptitude install maven2

7.1.3.2 Compilación Módulos

7.1.3.2.1 ImageSlideShow

Este módulo es un componente que permite incrustar un elemento para poder presentar imágenes en forma de slideshow.

- Para compilarlo inicialmente nos lo descargamos del svn del proyecto SourceForge

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer
shell>svn
https://cmosaic.svn.sourceforge.net/svnroot/cmosaic/Software/Modulos/ImageSlideShow
ImageSlideShow
shell>cd ImageSlideShow
shell>mvn package
shell>mvn install
shell>mvn deploy
```

7.1.3.2.2 MosaicModel

Este módulo define el modelo de objetos utilizados en la aplicación.

- Para compilarlo inicialmente nos lo descargamos del svn del proyecto SourceForge ...

```

shell>cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer

shell>svn
https://cmosaic.svn.sourceforge.net/svnroot/cmosaic/Software/Modulos/MosaicModel MosaicModel co

shell>cd MosaicModel

shell>mvn package

shell>mvn install

shell>mvn deploy

```

7.1.3.2.3 MosaicImageComposer-Server

Este módulo es el core de la aplicación.

- Para compilarlo inicialmente nos lo descargamos del svn del proyecto SourceForge.

```

shell>cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer

shell>svn
https://cmosaic.svn.sourceforge.net/svnroot/cmosaic/Software/Core/MosaicImageComposer-Server MosaicImageComposer-Server co

shell>cd MosaicImageComposer-Server

shell>mvn clean package

```

- Esto generará un directorio **/target/MosaicImageComposer-1.0** que posteriormente utilizaremos para generar el fichero WAR destino.

7.1.3.2.4 MosaicImageComposer-Client

Este módulo es el core de la aplicación.

- Para compilarlo inicialmente nos lo descargamos del svn del proyecto SourceForge.

```

shell>cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer

shell>svn
https://cmosaic.svn.sourceforge.net/svnroot/cmosaic/Software/Core/MosaicImageComposer-Client MosaicImageComposer-Client co

shell>cd MosaicImageComposer-Client

shell>mvn gwt:compile

```

- Esto generará un directorio **/target/MosaicImageComposer-1.0/com.ciclope.mosaicos.MosaicImageComposer** que posteriormente utilizaremos para generar el fichero WAR destino.

7.1.3.3 Generación del WAR (Web Archive)

Una vez generada la aplicación el siguiente paso es generar el fichero WAR con todos los módulos para posteriormente distribuirlo.

```
shell>/opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer/MosaicImageComposer-Server/target/
```

- Nos situamos en el directorio donde se distribuye el servidor y copiamos los módulos clientes directamente en el directorio

```
shell>cp -pR /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer/MosaicImageComposer-Client/target/MosaicImageComposer-1.0/com.ciclope.mosaicos.MosaicImageComposer/*
MosaicImageComposer-1.0

shell>cd MosaicImageComposer-1.0

shell>jar cvf ../MosaicImageComposer-1.0.war .
```

- Con esto tendremos generado el fichero destino **MosaicImageComposer-1.0.war** que podremos distribuir para su instalación.

7.2 Manual de instalación

Diferenciamos en este apartado la instalación de productos específicos necesarios para la ejecución del software y la instalación de los aplicativos desarrollados para poder generar mosaicos.

Concretamente se describirán los siguientes módulos

- Instalación de los servicios necesarios para poder ejecutar la aplicación (Tomcat, MySQL,...).
- Instalación de los aplicativos desarrollados como parte del proyecto.
- Previo al test de la aplicación se realizará la carga de Álbumes.
- Como último paso se realizará un test del aplicativo.

7.2.1 Instalación de servicios

Una vez realizada la compilación de las librerías necesarias es preciso preparar el entorno para la ejecución del software. Los productos necesarios para ejecutar el sistema son los siguientes.

- Máquina virtual Jdk 1.5
- Servidor de aplicaciones Apache Tomcat
- Base de datos Mysql
- Servidor Web Apache Server (Se necesita compilar el software)

Para alguno de estos productos se descargará directamente el binario mientras que para el resto será necesario realizar una compilación a partir de los fuentes.

Todos los productos descargados se situarán en un directorio específico de fuentes para tenerlos totalmente controlados. En este proyecto se ha decidido situarlo en el siguiente directorio.

Directorio
/opt/MosaicApp/MosaicSoftware.src

Para poder trabajar con el usuario **mosaicos** y no tener que necesitar hasta ciertos momentos el usuario **root** cambiaremos los permisos del directorio para que el usuario mosaicos pueda escribir en el.

```
shell>cd /opt
```

```
shell>sudo chown -R mosaicos:mosaicos /opt/MosaicApp
```

7.2.1.0.1 Requisitos

Prevía a la instalación del software es necesario verificar que el sistema dispone de los siguientes paquetes.

Paquete	Comando de búsqueda/instalación
SSH-Server	sudo aptitude search ssh sudo aptitude install ssh

7.2.1.1 Instalación JDK

Este producto permite la ejecución de aplicaciones Java así como la compilación de código. Se va a descargar por tanto una versión completa del JDK de Sun.

Para instalarlo hay que descargar antes que nada el código del JDK que se encuentra disponible en <http://java.sun.com/javase/downloads/index.jsp>. En este caso descargaremos el binario de la versión de Linux 32 bits.

La versión descargada era la más estable en el momento de preparación del entorno de instalación.

Versión	Fichero
1.5.0_17	jdk1.5.0_17.tgz

El software descargado lo situaremos en el siguiente directorio previa a la descompresión del mismo **1.JDK**.

```
/opt/MosaicApp/MosaicSoftware.src/1.JDK
```

Una vez descargado lo descomprimiremos utilizando los siguientes comandos

```
shell>cd /opt/MosaicApp  
  
shell>gzip -cd /opt/MosaicApp/MosaicSoftware.src/1.JDK/jdk1.5.0_17.tgz |tar xvf -
```

Esto generará un directorio denominado **/opt/MosaicApp/jdk1.5.0_17** localización final de la máquina virtual.

7.2.1.2 Instalación Apache Tomcat

Este producto cubre el rol de servidor de aplicaciones de la plataforma, más concretamente el de

contenedor de servlets. Para instalarlo hay que descargar antes que nada el código de Tomcat que se encuentra disponible en **<http://jakarta.apache.org>**. En este caso descargaremos los binarios, ya que al tratarse de un programa que está escrito en Java los podremos usar en cualquier plataforma con una JVM.

La versión descargada era la más estable en el momento de preparación del entorno de instalación.

Versión	Fichero
6.0.18	apache-tomcat-6.0.18.tar.gz

El software descargado lo situaremos en el siguiente directorio previa a la descompresión del mismo
2.ApacheTomcat.

```
/opt/MosaicApp/MosaicSoftware.src/2.ApacheTomcat
```

7.2.1.2.1 Preparación

Una vez descargado lo descomprimiremos utilizando los siguientes comandos

```
shell>cd /opt/MosaicApp

shell>gzip -cd /opt/MosaicApp/MosaicSoftware.src/2.ApacheTomcat/apache-tomcat-6.0.18.tar.gz
|tar xvf -
```

Esto generará un directorio denominado **/opt/MosaicApp/apache-tomcat-6.0.18** localización final del producto.

7.2.1.2.2 Personalización

Una vez realizada la instalación del servidor los siguientes pasos van destinados a la personalización del servicio. Para ello se van a realizar las siguientes tareas.

- Modificación de la variable LD_LIBRARY_PATH para permitir el proceso de conversión.
- Inclusión del path de Albums en el fichero **server.xml**

Para el primer cambio vamos a modificar el fichero **/opt/MosaicApp/apache-tomcat-6.0.18/bin/catalina.sh** he incluir las siguiente líneas. Se incluyen las líneas que aparecen en negrita y subrayadas en la posición marcada.

```
shell>vi /opt/MosaicApp/apache-tomcat-6.0.18/bin/catalina.sh

#####
#
# Get standard environment variables
PRGDIR=`dirname "$PRG"`

JAVA_OPTS="$JAVA_OPTS -Djava.awt.headless=true"
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/autopano:
```



```
/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/hugin:/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/convert
export JAVA_HOME=/opt/MosaicApp/jdk1.5.0_17
```

Para el segundo cambio vamos a modificar el fichero /opt/MosaicApp/apache-tomcat-6.0.18/conf/server.xml e incluir las siguientes líneas al final del fichero. Se incluyen las líneas que aparecen en negrita y subrayadas en la posición marcada. Aprovechamos también y activamos el log de accesos del servidor que aparece comentado por defecto.

```
shell>vi /opt/MosaicApp/apache-tomcat-6.0.18/conf/server.xml

#####
#
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log." suffix=".txt" pattern="common" resolveHosts="false"/>

    <!-- DIRECTORIO DE RECURSOS-->
        <Context docBase="/opt/MosaicApp/MosaicAlbums"
            path="/MosaicAlbums"
            reloadable="true"
            workDir="/opt/MosaicApp/MosaicAlbums/work">

    </Context>
```

7.2.1.2.3 Autoarranque

Para arrancar el servicio se ha generado un script de arranque que inicia todos los valores necesarios para arrancar el producto como es la localización de la máquina virtual Java. Este script con nombre tfc.tomcat.sh se debe situar en el directorio /etc/init.d del equipo y preparar un enlace simbólico S98tfc.tomcat.sh en el directorio /etc/rc2.d para que se arranque en el proceso de inicialización del equipo.

```
shell>cp tfc.tomcat.sh /etc/init.d

shell>sudo chmod +x /etc/init.d/tfc.tomcat.sh

shell>cd /etc/rc2.d

shell>ln -s /etc/init.d/tfc.tomcat.sh S98tfc.tomcat.sh
```

El contenido del script es el siguiente.

```
#!/bin/sh
export JAVA_HOME=/opt/MosaicApp/jdk1.5.0_17/
do_start() {
    su - mosaicos -c "/opt/MosaicApp/apache-tomcat-6.0.18/bin/startup.sh"
}

do_stop() {
    kill `ps -ef |grep tomcat|grep -v grep|grep -v tail|awk '{print $2}'`
}
case "$1" in
    start)
        do_start
        ;;
    restart|reload|force-reload)
        echo "Error: argument '$1' not supported" >&2
```

```

        exit 3
        ;;
    stop)
        do_stop
        # No-op
        ;;
    *)
        echo "Usage: tfc.tomcat.sh [start|stop]" >&2
        exit 3
        ;;
    esac

    exit 0

```

En algunas instalaciones se ha comprobado que este script en conjunción con el script de arranque del tomcat no permite que se autoarranque automáticamente la aplicación con el sistema operativo para lo cual es necesario modificar el script **catalina.sh** localizado en el directorio **/opt/MosaicApp/apache-tomcat-6.0.18/bin** y sustituir una de las líneas por esta otra en la que se elimina el símbolo **&** indicando que el proceso se pone en background. El cambio se puede observar en negrita en la siguiente ilustración.

```

else
"$_RUNJAVA" $JAVA_OPTS "$LOGGING_CONFIG" $CATALINA_OPTS \
-Djava.endorsed.dirs="$JAVA_ENDORSED_DIRS" -classpath "$CLASSPATH" \
-Dcatalina.base="$CATALINA_BASE" \
-Dcatalina.home="$CATALINA_HOME" \
-Djava.io.tmpdir="$CATALINA_TMPDIR" \
org.apache.catalina.startup.Bootstrap "$@" start \
>> "$CATALINA_BASE"/logs/catalina.out 2>&1 &

if [ ! -z "$CATALINA_PID" ]; then
    echo $! > $CATALINA_PID
fi

```

```

else
"$_RUNJAVA" $JAVA_OPTS "$LOGGING_CONFIG" $CATALINA_OPTS \
-Djava.endorsed.dirs="$JAVA_ENDORSED_DIRS" -classpath "$CLASSPATH" \
-Dcatalina.base="$CATALINA_BASE" \
-Dcatalina.home="$CATALINA_HOME" \
-Djava.io.tmpdir="$CATALINA_TMPDIR" \
org.apache.catalina.startup.Bootstrap "$@" start \
>> "$CATALINA_BASE"/logs/catalina.out 2>&1

```

7.2.1.2.4 Datos Configuración

Los datos de configuración básicos del servicio serán los siguientes:

Parámetro	Valor
Directorio	<u>/opt/MosaicApp/apache-tomcat-6.0.18</u>
Puerto	<u>8080</u>

7.2.1.2.4.1 Arranque

```
Sudo /etc/init.d/tfc.tomcat.sh start
```

7.2.1.2.4.2 Parada

```
Sudo /etc/init.d/tfc.tomcat.sh stop
```

7.2.1.3 Instalación Mysql

Este producto cubre el rol de base de datos de la plataforma. Para instalarlo hay que descargar antes que nada el código que se encuentra disponible en <http://www.mysql.com>.

La versión descargada era la más estable en el momento de preparación del entorno de instalación.

Versión	Fichero
5.1.36	mysql-5.1.36-linux-i686-glibc23.tar.gz

El software descargado lo situaremos en el siguiente directorio previa a la descompresión del mismo

3.Mysql.

```
/opt/MosaicApp/MosaicSoftware.src/3.Mysql
```

7.2.1.3.1 Preparación

Una vez descargado lo descomprimiremos utilizando los siguientes comandos

```
shell>cd /opt/MosaicApp

shell>gzip -cd /opt/MosaicApp/MosaicSoftware.src/3.Mysql/mysql-5.1.36-linux-i686-
glibc23.tar.gz |tar xvf -
```

Esto generará un directorio denominado **/opt/MosaicApp/mysql-5.1.36-linux-i686-glibc23** localización del producto para su preparación. Seguidamente ejecutamos los siguientes comandos para dejar finalizada la instalación

```
shell>sudo groupadd mysql

shell>sudo useradd -g mysql mysql

shell>cd /usr/local

shell>sudo ln -s /opt/MosaicApp/mysql-5.1.36-linux-i686-glibc23 mysql

shell>cd /opt/MosaicApp/mysql-5.1.36-linux-i686-glibc23

shell>sudo chown -R mysql .

shell>sudo chgrp -R mysql .

shell>sudo scripts/mysql_install_db --user=mysql

shell>sudo chown -R root .

shell>sudo chown -R mysql data
```

7.2.1.3.2 Autoarranque

Para arrancar el servicio se ha generado un script de arranque que inicia todos los valores necesarios para arrancar el producto. Este script con nombre **tfc.mysql.sh** se debe de situar en el directorio **/etc/init.d** del equipo y preparar un enlace simbólico **S98tfc.tomcat.sh** en el directorio **/etc/rc2.d** para que se arranque en el proceso de inicialización del equipo.

```
shell>sudo cp tfc.tomcat.sh /etc/init.d

shell>sudo chmod +x /etc/init.d/tfc.mysql.sh

shell>cd /etc/rc2.d

shell>sudo ln -s /etc/init.d/tfc.mysql.sh S98tfc.mysql.sh
```

El contenido del script es el siguiente.

```
#!/bin/sh
do_start() {
    /usr/local/mysql/bin/mysqld_safe --user=mysql &
}
do_stop() {
    kill `ps -ef |grep mysqld|grep -v grep|awk '{print $2}'`
    kill -9 `ps -ef |grep mysqld_safe|grep -v grep|awk '{print $2}'`
}

case "$1" in
    start)
        do_start
        ;;
    restart|reload|force-reload)
        echo "Error: argument '$1' not supported" >&2
        exit 3
        ;;
    stop)
        do_stop
        # No-op
        ;;
    *)
        echo "Usage: tfc.mysql.sh [start|stop]" >&2
        exit 3
        ;;
esac

exit 0
```

7.2.1.3.3 Datos Configuración

Los datos de configuración básicos del servicio serán los siguientes:

Parámetro	Valor
Directorio	<u>/opt/MosaicApp/mysql-5.1.36-linux-i686-glibc23</u>
Puerto	<u>3306</u>

7.2.1.3.3.1 Arranque

```
sudo /etc/init.d/tfc.mysql.sh start
```

7.2.1.3.3.2 Parada

```
sudo /etc/init.d/tfc.mysql.sh stop
```

7.2.1.4 Instalación Apache Httpd

Este producto cubre el rol de base de servidor web de la plataforma. Para instalarlo hay que descargar antes que nada el código que se encuentra disponible en **<http://httpd.apache.org>**.

La versión descargada era la más estable en el momento de preparación del entorno de instalación.

Versión	Fichero
2.2.14	httpd-2.2.14.tar.gz

El software descargado lo situaremos en el siguiente directorio previa a la descompresión del mismo **4.ApacheHttpd**.

```
/opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd
```

7.2.1.4.1 Preparación

Una vez descargado lo descomprimiremos utilizando los siguientes comandos. El software de apache que nos descargamos necesita ser compilado para su uso

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd  
  
shell>gzip -cd /opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd/httpd-2.2.14.tar.gz |tar xvf -
```

Esto generará un directorio denominado **/opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd/httpd-2.2.14** localización del producto para su compilación.

Los siguientes pasos permiten compilar el producto y localizarlo el producto en el directorio destino que será el siguiente **/opt/MosaicApp/httpd-2.2.14**

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd/httpd-2.2.14  
  
shell>./configure --prefix=/opt/MosaicApp/httpd-2.2.14 --enable-module=so  
  
shell>make  
  
shell>make install
```

7.2.1.4.2 Personalización

Una vez realizada la compilación e instalación del servidor los siguientes pasos van destinados a la personalización del servicio. Para ello se van a realizar las siguientes tareas.

- Cambio del puerto donde arranca el servidor web que por defecto es el 80

Para cambiar el puerto editaremos el fichero de configuración /opt/MosaicApp/httpd-2.2.14/conf/httpd.conf y modificaremos el parámetro Listen por si existe algún otro servicio que escuche en el puerto 80, puerto típico en las instalaciones de servidor Web. Pondremos en su lugar en valor 9090.

```
shell>vi /opt/MosaicApp/httpd-2.2.14/conf/httpd.conf

#####
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 9090
#####
```

7.2.1.4.3 Autoarranque

Para arrancar el servicio se ha generado un script de arranque que inicia todos los valores necesarios para arrancar el producto. Este script con nombre tfc.httpd.sh se debe de situar en el directorio /etc/init.d del equipo y preparar un enlace simbólico S98tfc.httpd.sh en el directorio /etc/rc2.d para que se arranque en el proceso de inicialización del equipo.

```
shell>sudo cp tfc.httpd.sh /etc/init.d

shell>sudo chmod +x /etc/init.d/tfc.httpd.sh

shell>cd /etc/rc2.d

shell>sudo ln -s /etc/init.d/tfc.httpd.sh S98tfc.httpd.sh
```

El contenido del script es el siguiente.

```
#!/bin/sh
do_start() {
    su - mosaicos -c "/opt/MosaicApp/httpd-2.2.14/bin/apachectl start"
}

do_stop() {
    su - mosaicos -c "/opt/MosaicApp/httpd-2.2.14/bin/apachectl stop"
}
case "$1" in
    start)
        do_start
        ;;
    restart|reload|force-reload)
        echo "Error: argument '$1' not supported" >&2
        exit 3
        ;;
    *)
```

```
stop)
do_stop
# No-op
;;
*)
echo "Usage: tfc.httpd.sh [start|stop]" >&2
exit 3
;;
esac
exit 0~
```

7.2.1.4.4 Datos Configuración

Los datos de configuración básicos del servicio serán los siguientes:

Parámetro	Valor
Directorio	<u>/opt/MosaicApp/httpd-2.2.14</u>
Puerto	9090 (El valor por defecto es el 80)

7.2.1.4.4.1 Arranque

```
sudo /etc/init.d/tfc.httpd.sh start
```

7.2.1.4.4.2 Parada

```
sudo /etc/init.d/tfc.httpd.sh stop
```

7.2.2 Instalación de aplicativos

7.2.2.1 Plugins Apache

El siguiente paso es la instalación de módulos adicionales (plugins) para el soporte de fast-cgi y proxy. En apache los módulos se deben de instalar independientemente y permiten ampliar las funcionalidades del producto.

7.2.2.1.1 Fast-Cgi

Para instalarlo hay que descargar antes que nada el código que se encuentra disponible en <http://www.fastcgi.com>.

La versión descargada era la más estable en el momento de preparación del entorno de instalación.

Versión	Fichero
2.4.6	mod_fastcgi-2.4.6.tar.gz

El software descargado lo situaremos en el siguiente directorio previa a la descompresión del mismo

4.ApacheHttpd.

```
/opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd/modules
```

7.2.2.1.1.1 Preparación

Una vez descargado lo descomprimiremos utilizando los siguientes comandos. El software de apache que nos descargamos necesita ser compilado para su uso

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd/modules

shell>gzip -cd /opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd/modules/mod_fastcgi-2.4.6.tar.gz |tar xvf -
```

Esto generará un directorio denominado /opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd/modules/mod_fastcgi-2.4.6 localización del producto para su compilación.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd/modules/mod_fastcgi-2.4.6

shell>cp Makefile.AP2 Makefile

shell>make top_dir=/opt/MosaicApp/httpd-2.2.14

shell>make install top_dir=/opt/MosaicApp/httpd-2.2.14
```

7.2.2.1.2 Mod-Proxy

El software de apache proxy viene con la distribución del software httpd por lo que no es necesario realizar ninguna descarga de software adicional.

7.2.2.1.2.1 Preparación

Como previamente se ha instalado el software de servidor Web nos cambiamos al directorio generado y localizamos el modulo de proxy.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd/httpd-2.2.14/modules/proxy
```

Procedemos a continuación con la compilación.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/4.ApacheHttpd/httpd-2.2.14/modules/proxy

Shell>/opt/MosaicApp/httpd-2.2.14/bin/apxs -i -a -c mod_proxy.c proxy_util.c

shell>/opt/MosaicApp/httpd-2.2.14/bin/apxs -i -n mod_proxy -c mod_proxy.c proxy_util.c

shell>/opt/MosaicApp/httpd-2.2.14/bin/apxs -i -n mod_proxy -c mod_proxy_http.c proxy_util.c

shell>/opt/MosaicApp/httpd-2.2.14/bin/apxs -i -n mod_proxy -c mod_proxy_ajp.c proxy_util.c

shell>/opt/MosaicApp/httpd-2.2.14/bin/apxs -i -n mod_proxy -c mod_proxy_balancer.c proxy_util.c

shell>/opt/MosaicApp/httpd-2.2.14/bin/apxs -i -n mod_proxy -c mod_proxy_connect.c proxy_util.c

shell>/opt/MosaicApp/httpd-2.2.14/bin/apxs -i -n mod_proxy -c mod_proxy_ftp.c proxy_util.c
```

7.2.2.1.3 Iipmooviewer

Este software es un sistema cliente / servidor para la visualización y la navegación dentro de muy alta resolución de imágenes. Se basa en el Protocolo de Internet de imágenes (iiop) y es capaz de trabajar de forma eficiente, incluso con conexiones de ancho de banda reducida.

El servidor está escrito en C++ como un plugin que se puede instalar en diferentes tipos de servidores como Apache, lighttpd, o cualquier otro software que soporta fast-cgi (FCGI). Las imágenes se visualizan a través de un Javascript AJAX, Flash o Java cliente. El sistema puede también manejar las secuencias de imagen en 3D, panorámicas o imágenes multiespectrales. El software dispone de Licencia: GNU General Public License (GPL).

El software se puede descargar de la siguiente dirección de Internet <http://iipimage.sourceforge.net/documentation/iipmooviewer/>

La versión descargada era la más estable en el momento de preparación del entorno de instalación. Es necesario una versión de cliente y una versión de servidor.

NOTA: Para el software de cliente a partir de la versión descargada de Internet se ha generado un adaptación de la misma que se proporciona como parte del proyecto y permite que pueda ser integrada dentro de un servidor Tomcat ya que se incluye soporte “JSP” al mismo con el objetivo de que pueda ser utilizado genéricamente dentro de una aplicación. Por tanto la distribución que se proporcionará será la adaptada.

Para el software de servidor se ha realizado también una adaptación ya que no era compatible con el software de compilación C++ de la distribución ubuntu.

Versión	Fichero
0.9.7 (Servidor-Original)	iipsrv-0.9.7.tar.bz2
0.9.7 (Servidor-Adaptado)	iipsrv-adapted-0.9.7.tar.bz2
1.1 (Cliente AJAX original)	iipmooviewer-1.1.tar.gz.
1.1 (Cliente AJAX adaptado)	iipmooviewer-adapted-1.1.tar.gz.

El software descargado lo situaremos en el siguiente directorio previa a la descompresión del mismo **5.Iipmooviewer.**

```
/opt/MosaicApp/MosaicSoftware.src/5.Iipmooviewer
```

7.2.2.1.4 Iipmooviewer-server

7.2.2.1.4.1 Requisitos

Prevía a la instalación de la librería es necesario disponer de al menos los siguientes paquetes.

Paquete	Comando búsqueda/instalación
libjpeg	sudo aptitude search libjpeg sudo aptitude install libjpeg62 sudo aptitude install libjpeg-dev
Libtiff	sudo aptitude search libtiff sudo aptitude install libtiff-dev
G++ (Compilador de C++)	sudo aptitude search build-essential sudo aptitude install build-essential

7.2.2.1.4.2 Preparación

Una vez descargado lo descomprimiremos utilizando los siguientes comandos. El software de servidor que nos descargamos necesita ser compilado para su uso.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/5.Iipmooviewer  
  
shell>bzcat /opt/MosaicApp/MosaicSoftware.src/5.Iipmooviewer/iipsrv-adapted-0.9.7.tar.bz2 |  
tar xvf -
```

Una vez descomprimido pasamos a realizar la compilación y la generación del fichero fast-cgi.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/5.Iipmooviewer/iipsrv-0.9.7  
  
shell>./configure--prefix=/opt/MosaicApp/iipsrv-0.9.7  
  
shell>make  
  
shell>make install
```

Seguidamente copiamos el fichero fast-cgi generado en el servidor web.

```
shell>mkdir /opt/MosaicApp/httpd-2.2.14/cgi-bin  
  
shell>cp src/iipsrv.fcgi /opt/MosaicApp/httpd-2.2.14/cgi-bin
```

El siguiente paso es la modificación del fichero de configuración del servidor web para configurar el soporte fast-cgi. Modificamos el fichero de configuración /opt/MosaicApp/httpd-2.2.14/conf/httpd.conf y añadimos al final del fichero las siguientes sentencias.

```
shell>vi /opt/MosaicApp/httpd-2.2.14/conf/httpd.conf
```

```
#####
# Configuración Mosaicos
Include conf/extra/httpd-mosaicos.conf
# Fin configuracion Mosaicos
#####
```

Añadimos un fichero denominado **/opt/MosaicApp/httpd-2.2.14/conf/extra/https-mosaicos.conf**. Este fichero almacena la configuración de soporte fast-cgi.

```
shell>vi /opt/MosaicApp/httpd-2.2.14/conf/extra(https-mosaicos.conf

#####
##### CONTENIDO DEL FICHERO#####
#####
<IfModule !mod_fastcgi.c>
    LoadModule fastcgi_module modules/mod_fastcgi.so
</IfModule>

<IfModule !mod_proxy.c>
    LoadModule proxy_module modules/mod_proxy.so
</IfModule>

<IfModule mod_proxy.c>
    LoadModule proxy_http_module modules/mod_proxy_http.so
    LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
    LoadModule proxy_connect_module modules/mod_proxy_connect.so
</IfModule>

<IfModule mod_fastcgi.c>
# Initialise some variables for the FCGI server
FastCgiServer /opt/MosaicApp/httpd-2.2.14/fcgi-bin/iipsrv.fcgi \
-initial-env LOGFILE=/tmp/iipsrv.log \
-initial-env VERBOSITY=2 \
-initial-env MAX_IMAGE_CACHE_SIZE=10 \
-initial-env FILENAME_PATTERN=_pyr_ \
-initial-env JPEG_QUALITY=50 \
-initial-env MAX_CVT=3000 \
-initial-env LD_LIBRARY_PATH=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/iipsrv
</IfModule>
<IfModule alias_module>
    ScriptAlias /fcgi-bin/ "/opt/MosaicApp/httpd-2.2.14/fcgi-bin/"
</IfModule>

# Set the options on that directory
<Directory "/opt/MosaicApp/httpd-2.2.14/fcgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

#ProxyPass /fcgi-bin/iipsrv.fcgi http://localhost/fcgi-bin/iipsrv.fcgi
#ProxyPassReverse /fcgi-bin/iipsrv.fcgi http://remotehost/fcgi-bin/iipsrv.fcgi
```

```
ProxyPass /iipmooviewer http://localhost:8080/iipmooviewer
ProxyPassReverse /iipmooviewer http://localhost:8080/iipmooviewer

ProxyPass /com.tfc.mosaicos.MosaicImageComposer http://localhost:8080/com.tfc.mosaicos.MosaicImageComposer
ProxyPassReverse /com.tfc.mosaicos.MosaicImageComposer http://localhost:8080/com.tfc.mosaicos.MosaicImageComposer

ProxyPass /MosaicAlbums http://localhost:8080/MosaicAlbums
ProxyPassReverse /MosaicAlbums http://localhost:8080/MosaicAlbums

ProxyPass /MosaicImageComposer http://localhost:8080/MosaicImageComposer-1.0
ProxyPassReverse /MosaicImageComposer http://localhost:8080/MosaicImageComposer-1.0
```

7.2.2.1.5 Iipmooviewer-client

7.2.2.1.5.1 Preparación

Una vez descargado lo descomprimiremos utilizando los siguientes comandos.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/5.Iipmooviewer

shell>gzip -cd /opt/MosaicApp/MosaicSoftware.src/5.Iipmooviewer/iipmooviewer-adapted-1.1.tar.gz |tar xvf -
```

Una vez descomprimido lo movemos al directorio webapps del tomcat.

```
shell>mv iipmooviewer /opt/MosaicApp/apache-tomcat-6.0.18/webapps/
```

7.2.2.2 Software Mosaicos

En este apartado se muestra el proceso de instalación del software de mosaicos.

La versión instalada corresponde a la desarrollada como parte de este proyecto cuyo proceso de generación se puede ver en el apartado 7.1 en el que se explica el proceso de compilación del software.

Se distribuyen tres tipos de componentes el software de interfaz de usuario gwt, las librerías de mosaicos generadas y las librerías de productos necesarios para la generación de mosaicos librerías utilizadas por el producto hugin.

Versión	Fichero
1.0	MosaicImageComposer-1.0.tar.gz
1.1	libMosaic-1.1.tar.gz
1.0	libMosaicProducts-1.1.tar.gz

El software descargado lo situaremos en el siguiente directorio previa a la descompresión del mismo **6.MosaicImageComposer**.

```
/opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer
```

7.2.2.2.1 Requisitos

Prevía a la instalación de los productos necesario disponer de al menos los siguientes paquetes.

Paquete	Comando búsqueda/instalación
Imagemagick (convert command)	sudo aptitude search imagemagick sudo aptitude install imagemagick
Vips (Image Processing Tool) (Vips command)	sudo aptitude search vips sudo aptitude install libvips15 sudo aptitude install libvips-tools
Exiftool	sudo aptitude search libimage-exiftool-perl sudo aptitude install libimage-exiftool-perl

7.2.2.2.2 Instalación Productos

7.2.2.2.2.1 MosaicImageComposer

Una vez descargado lo descomprimiremos utilizando los siguientes comandos.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer
```

```
shell>gzip -cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer/MosaicImageComposer-1.0.tar.gz |tar xvf -
```

Esto generará un directorio denominado **/opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer/MosaicImageComposer-1.0**

En el siguiente paso movemos este directorio a la localización del directorio de aplicaciones del tomcat.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer  
  
shell>mv MosaicImageComposer-1.0 /opt/MosaicApp/apache-tomcat-6.0.18/webapps/
```

7.2.2.2.2 libMosaic

Una vez descargado lo descomprimiremos utilizando los siguientes comandos.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer  
  
shell>gzip -cd libMosaic-1.1.tar.gz |tar xvf -
```

Esto generará un directorio denominado **/opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer/libMosaic**

En el siguiente paso movemos este directorio a la localización del directorio de aplicaciones del tomcat.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer  
  
shell>mkdir /opt/MosaicApp/MosaicImageComposer  
  
shell>mv libMosaic /opt/MosaicApp/MosaicImageComposer
```

7.2.2.2.3 libProducts

Una vez descargado lo descomprimiremos utilizando los siguientes comandos.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer  
  
shell>gzip -cd libMosaicProducts-1.1.tar.gz |tar xvf -
```

Esto generará un directorio denominado **/opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer/libMosaicProducts**

En el siguiente paso movemos este directorio a la localización del directorio de aplicaciones del tomcat.

```
shell>cd /opt/MosaicApp/MosaicSoftware.src/6.MosaicImageComposer  
  
shell>mkdir /opt/MosaicApp/MosaicImageComposer
```

```
shell>mv libMosaicProducts /opt/MosaicApp/MosaicImageComposer/
```

7.2.2.2.3 Configuración

La personalización de los productos de mosaicos se lleva a cabo mediante la modificación del fichero de configuración de la aplicación. Este fichero se localiza en el directorio /opt/MosaicApp/MosaicImageComposer con nombre mosaic.properties

```
shell>vi /opt/MosaicApp/MosaicImageComposer/mosaic.properties

#####

#Localizacion de los ficheros generados
app.mosaic.directory.albums=/opt/MosaicApp/MosaicAlbums/
app.mosaic.directory.generated=/opt/MosaicApp/MosaicAlbums/generated/
app.mosaic.directory.tempgenerated=/tmp/generated/

#Comando para generar los puntos de interes
app.mosaic.command.autopano.name=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/autopano/autopano-sift-c.sh
app.mosaic.command.autopano.points=20

#Cada cuanto tiempo el sistema chequea los procesos caducados. Segundos
app.mosaic.process.timercaducados=60
#Máximo tiempo de vida que un proceso permanece en el sistema. Segundos
app.mosaic.process.maxlifecaducados=3600

#Conversor de imagenes
app.mosaic.command.vips.name=/usr/bin/vips
app.mosaic.command.vips.parameters=:jpeg:75,tile:256x256,pyramid

#Programas para realizar la conversión
app.mosaic.command.nona.name=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/panorama/nona
app.mosaic.command.hdrmerge.name=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/panorama/hugin_hdrmerge
app.mosaic.command.PTmender.name=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/panorama/PTmender
app.mosaic.command.PTblender.name=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/panorama/PTblender
app.mosaic.command.PTmasker.name=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/panorama/PTmasker
app.mosaic.command.PTroller.name=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/panorama/PTroller
app.mosaic.command.enblend.name=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/panorama/enblend
app.mosaic.command.enblend.args=

app.mosaic.command.enfuse.name=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/panorama/enfuse
app.mosaic.command.enfuse.args=
app.mosaic.command.exiftool.name=/usr/bin/exiftool
app.mosaic.command.exiftool.args=

app.mosaic.command.make.name=/opt/MosaicApp/MosaicImageComposer/libMosaicProducts/panorama/make.sh

#Programa para la eliminación del canal alpha de las imagenes
app.mosaic.command.imagemagickconvert.name=/usr/bin/convert
app.mosaic.command.imagemagickconvert.parameters.alpha=-alpha Off
app.mosaic.command.imagemagickconvert.parameters.rgb=-type TrueColor
```

Los valores de configuración de este fichero y su descripción son los siguientes:

Campo	Descripción
app.mosaic.directory.albums	Directorio físico donde se localizan los álbumes de los usuarios
app.mosaic.directory.generated	Directorio donde se generan las imágenes temporales
app.mosaic.directory.tempgenerated	Directorio donde se generan las imágenes temporales
app.mosaic.command.autopano.name	Localización del comando autopano-sift-c. En este caso se indica la localización de un script que inicializa una serie de valores previos a la ejecución del comando autopano-sift-c
app.mosaic.command.autopano.points	Número de puntos de control a generar por el comando autopano.
app.mosaic.process.timercaducados	Cada cuanto tiempo en segundos se chequea los procesos existentes.
app.mosaic.process.maxlifecaducados	Máximo tiempo de vida que un proceso interno de generación de mosaicos permanece activo
app.mosaic.command.vips.name	Localización del programa conversor de imágenes.
app.mosaic.command.vips.parameters	Parámetros de conversión
app.mosaic.command.nona.name	Localización del programa nona
app.mosaic.command.hdrmerge.name	Localización del programa hdrmerge de la suite PanoramaTools
app.mosaic.command.PTMender.name	Localización del programa PTMender de la suite PanoramaTools
app.mosaic.command.PTblender.name	Localización del programa PTBlender de la suite PanoramaTools
app.mosaic.command.PTmasker.name	Localización del programa PTMasker de la suite PanoramaTools
app.mosaic.command.PTRoller.name	Localización del programa PTRoller de la suite PanoramaTools
app.mosaic.command.enblend.name	Localización del programa enblend de la suite PanoramaTools
app.mosaic.command.enblend.args	Argumentos para el programa enblend de la suite PanoramaTools
app.mosaic.command.enfuse.name	Localización del programa enfuse de la suite PanoramaTools
app.mosaic.command.enfuse.args	Argumentos para el programa enfuse
app.mosaic.command.exiftool.name	Localización del programa exiftool de la suite PanoramaTools
app.mosaic.command.exiftool.args	Argumentos para el programa exiftool
app.mosaic.command.make.name	Localización del programa make de la suite PanoramaTools
app.mosaic.command.imagemagickconvert.name	Localización del programa convert de la suite ImageMagick
app.mosaic.command.imagemagickconvert.parameters.alpha	Parámetros alpha del programa convert de la suite ImageMagick
app.mosaic.command.imagemagickconvert.parameters.rgb	Parámetros rgb del programa convert de la suite ImageMagick

7.2.2.2.4 Carga de datos

Una vez modificados estos parámetros el siguiente paso será cargar el script sql de la base de datos con toda la información.

Para ello podemos utilizar cualquier herramienta para conectarse a la Base de Datos por ejemplo una herramienta gráfica tipo MYSQL Administrator (www.mysql.com) o las propias herramientas proporcionados por la distribución de mysql desplegada.

En primer lugar es preciso configurar el equipo para que se pueda acceder desde varios equipos

```
shell>/usr/local/mysql/bin/mysql -u root -p
mysql><Password de root Inicialmente vaciá>
mysql>create database mosaicos;
mysql>show databases;
shell>/usr/local/mysql/bin/mysql -u root -p
mysql>grant all on mysql.* to root@'%' identified by '';
mysql>grant all on mosaicos.* to root@'%' identified by '';
```

Seguidamente cargamos la base de Datos con las tablas y datos necesarios.

```
shell>/usr/local/mysql/bin/mysql -h localhost -u root mosaicos <
/opt/MosaicApp/MosaicSoftware.src/3.MySql/BackupMosaicos.sql
```

7.2.3 Carga de Albums

Si disponemos de información sobre Álbumes previamente creados deberemos crear la estructura para almacenar en el directorio **MosaicAlbums**

```
shell>mkdir -p /opt/MosaicApp/MosaicAlbum  
  
shell>mkdir -p /opt/MosaicApp/MosaicAlbum/generated  
  
shell>mkdir -p /opt/MosaicApp/MosaicAlbum/work
```

7.2.4 Test de aplicativos

Previo a realizar el test del aplicativo se recomienda realizar una parada del servidor de aplicaciones (Tomcat) y del servidor Web (httpd).

```
shell>sudo /etc/init.d/tfc.httpd.sh stop  
  
shell>sudo /etc/init.d/tfc.httpd.sh start  
  
shell>sudo /etc/init.d/tfc.tomcat.sh stop  
  
shell>sudo /etc/init.d/tfc.tomcat.sh start
```

Para verificar el correcto funcionamiento de la aplicación introducimos en el navegador la siguiente URL.

<http://mosaicos:9090/MosaicImageComposer/MosaicImageComposer.html>

Donde el nombre “mosaicos” hace referencia a la dirección IP del equipo donde se ha instalado el equipo.

8 Conclusiones y líneas futuras

Este último capítulo pretende presentar de forma breve los puntos fundamentales del trabajo desarrollado, exponer las principales conclusiones obtenidas y presentar las líneas de investigación que quedan abiertas.

8.1 Conclusiones

El objetivo principal de este proyecto ha sido la realización de un módulo web que permitiera a los usuarios poder generar imágenes panorámicas también conocidas como mosaicos, tomadas bien a través del telescopio del proyecto o mediante la carga de las mismas en la aplicación desde el ordenador.

La principal conclusión que se ha obtenido en el desarrollo del trabajo es que existe una gran cantidad de material en Internet y muchos foros dedicado a técnicas de composición de imágenes y más concretamente a la faceta matemática de estas técnicas.

El principal problema de la composición de mosaicos según se ha podido estudiar es la calidad y nitidez de las imágenes que se toman y que afectan al proceso global de composición. Cuando la calidad, el contraste o la luminosidad de las mismas no es la adecuada el proceso de composición se resiente. Es en este momento donde entran en juego algoritmos matemáticos y filtros para el procesamiento de imágenes que intentan solucionar estos problemas.

También es importante durante el proceso de toma de las imágenes que en todas las imágenes que se generen se trabaje con los mismos valores de velocidad de obturación, apertura de diafragma, etc. En la mayoría de las cámaras compactas del mercado el modo automático es el por defecto y es el principal enemigo de los mosaicos. No obstante existen algoritmos que pretenden mitigar estos problemas con unos resultados muy aceptables.

Uno de los grandes retos del proyecto ha sido las tareas de integración de los distintos componentes utilizados en el proyecto. Por una parte la interfaz GWT, la interfaz de servidor y la librería dinámica de composición de mosaicos. Uno de los principales objetivos de la integración ha sido diseñar la misma lo suficientemente modular para realizar futuras ampliaciones del sistema.

Como conclusión final podemos decir que se ha implementado un sistema de composición de mosaicos plenamente operativo válido en principio para cualquier tipo de imagen con las características que se han comentado anteriormente, cuyos resultados son bastante adecuados a nivel de usuario.

8.2 Líneas futuras

El trabajo efectuado ha dejado líneas abiertas para posibles investigaciones futuras. A continuación se comentan las que parecen más interesantes.

- Los algoritmos de detección de características para obtener las similitudes entre las imágenes y

poder realizar la fusión de las mismas están evolucionado principalmente en dos aspectos, el primero intentar ser más precisos a la hora de obtener puntos representativos en las imágenes ante mientras que por otra parte se está intentando agilizar la búsqueda de características. En este proyecto se ha trabajado con el algoritmo SIFT aunque existen otros algoritmos de detección como SURF que puede ser interesante su inclusión dentro del proyecto.

- Otro línea de futuro es la generación y manejo de imágenes en formato QTVR (Quick Time) para poder trabajar con imágenes en formato 360° (Panorámicas esféricas). No se ha trabajado con estas imágenes en este proyecto al centrarse el mismo en imágenes de astrofotografía, pero una amplia mayoría de composiciones disponibles en Internet están generadas en este espectacular formato.
- La interfaz de usuario es otro punto de evolución futura en el sistema. La interfaz que se ha generado como producto final pretende ser un asistente sencillo a la hora de generar un mosaico. No obstante el proceso de composición de imágenes es algo más complicado cuando las imágenes tienen problemas de luminosidad, contraste o tienen defectos para los cuales es necesario realizar operaciones más manuales.
 - Como mejoras de esta interfaz se propone la inclusión de un editor de puntos que pretende incluir, borrar o desplazar de forma manual los puntos que identifican características similares entre las imágenes. El algoritmo de obtención automático en algunos casos puede ser no demasiado preciso y puede ser interesante introducir el componente humano en el sistema.
 - Inclusión de más campos para poder modificar más parámetros de la imagen.
- Potenciar la compartición de información entre los usuarios mediante la inclusión de alguna herramienta que permita incluir comentarios sobre las imágenes, sobre como se han tomado, en que momento y como se ha realizado la composición para ayudar a otros usuarios a generar imágenes con un nivel de calidad aceptable. La inclusión de un enlace a una aplicativo tipo blog puede ser muy interesante para potenciar la herramienta.

Bibliografía

Bibliografía

SIFT: David G. Lowe, Object recognition from local scale-invariant features, 1999
SURF: Herbert Bay, T. Tuytelaars and L. Van Gool, SURF: Speeded Up Robust Features, 2006
VIGNETTING: , VIÑETEO - Nociones Básicas, ,
<http://www.digitalcamaralens.com/Html/Articulos/Vineteo/Vineteo.htm>
CELESTE: , , <http://ultrawide.wordpress.com/2008/11/12/using-celeste/>
CICLOPE09: , Página del proyecto Cíclope, 2009,
CICLOPE07: R. CEDAZO; D. LOPEZ; F. M. SANCHEZ; J. M. SEBASTIAN, Ciclope: FOSS for
Developing and Managing Educational Web Laboratories, November 2007
HUGIN: , Software Hugin, 2009, <http://hugin.sourceforge.net>
PANOTOOLS: , Panorama Tools, , <http://www.panotools.org>
WXWIDGETS: , , 2009, <http://www.wxwidgets.org/>
HUGINSRC: , , 2009, <http://hugin.sourceforge.net/download/>
ENFUSE: T. Mertens, J. Kautz and F. Van Reeth, Exposure Fusion, 2007
W3C: , W3C, , <http://www.w3.org>
AULAMBRA: , Estadísticas de uso de navegadores, 2009,
<http://aulambra.lineadecodigo.com/tag/internet-explorer/>
GWT: , GWT, , <http://code.google.com/intl/es-ES/webtoolkit/>

9 Apéndice 1. Presupuesto.

En este apartado se realiza un resumen de los recursos humanos y tecnológicos involucrados en el desarrollo del proyecto.

9.1 Fases

Las distintas fases del proyecto y su duración se pueden ver en la siguiente tabla.

Fase	Duración (Días)	Descripción
Reunión Inicio Proyecto	1	Se realiza la reunión de inicio del proyecto.
Estudio alternativas	40	Se revisan el estado del arte en las aplicaciones de tipo mosaicos revisando las alternativas de software disponibles en el mercado.
Análisis y Diseño	20	Se realiza el análisis y diseño de la aplicación
Coordinación Desarrollo	62	La coordinación del desarrollo por parte del analista permite disponer de un software de calidad.
Diseño Interfaces web	10	Se diseñan el interfaz web de la aplicación.
Diseño Interfaz C++	20	Se realiza el desarrollo del módulo de servidor
Diseño Interfaz GWT	32	Se realiza el desarrollo del módulo cliente y la interconexión con el módulo servidor.
Documentación	53	Documentación del proyecto.
Resolución Incidencias	55	Tras las pruebas realizadas por los integrantes del grupo de mosaicos se van resolviendo las incidencias.
Preparación Formación	20	Se prepara la formación del proyecto.

9.2 Recursos.

Recurso	Iniciales	Capacidad Máxima	Coste
---------	-----------	------------------	-------

Senior Project Manager	SPM	30,00%	43,9
Senior Engineer	SE	75,00%	32,1
Senior Technician	ST	100,00%	21,49
Graphic Designer	GD	100,00%	21,49

9.3 Recursos tecnológicos.

Para la realización de este proyecto no se ha necesitado ningún paquete de software adicional ya que se ha intentado utilizar tanto sistemas operativos tipo Linux (Ubuntu) como aplicaciones ofimática sin coste (OpenOffice)

9.4 Presupuesto de Costes.

En la Figura Figura 118 se puede ver el presupuesto de costes del proyecto con las horas de los recursos involucrados en el proyecto.

	Senior Project Manager	Senior Engineer	Senior Technician	Senior Technician	Total fase €
Fase 1	187	389	388	80	30.752,67
Fase 2					0,00
Fase 3					0,00

Figura 118: Presupuesto de costes

En un proyecto real es importante realizar una valoración extra con el fin de conseguir beneficios adicionales con el fin de no solo cubrir costes. Para ello es preciso incorporar un margen de venta al proyecto. En este caso hemos incluido un margen de un 25% sobre costes quedando el presupuesto de venta final reflejado en la figura Figura 119.

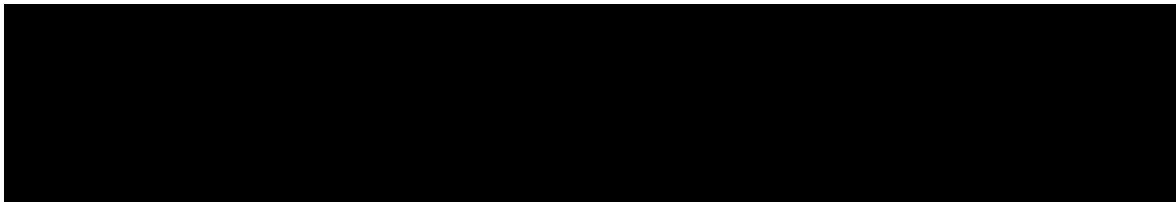


Figura 119: Presupuesto de venta

10 Apéndice 2. Fotografía digital

En este apartado se realiza un resumen de aquellas características en la fotografía digital que pueden ser útiles para conocer el trabajo realizado.

10.1 Conceptos Básicos

10.1.1 Distancia Focal

El objetivo es la parte de la cámara fotográfica que más directamente tiene que ver con la imagen que va a conseguirse. La característica más notable de un objetivo es su **distancia focal**. Se pueden emplear objetivos con distintas distancias focales (28 mm, 50 mm 135 mm, etc.) y ello determina por una parte el tamaño de la imagen y por otro la profundidad de campo. La distancia focal de una lente es la distancia entre el centro óptico de la lente y el foco (o punto focal). El foco es el punto donde se concentran los rayos de luz.

El ángulo de visión de un objetivo depende de su distancia focal. Los objetivos de 50 mm son los más habituales porque ofrecen una imagen parecida a la del ojo humano. Los de distancia focal menor (p.e.: 28 mm) tienen un mayor ángulo de visión y permiten captar un área muy grande de la escena (son empleados en la fotografía paisajística, entre otros). Los de distancia focal mayor (p.e.: 200 mm) tienen un ángulo de visión menor y captan una parte más pequeña de la escena. Con estos últimos la imagen parece ampliada o tomada desde muy cerca (se emplean en la fotografía de primeros planos).

10.1.2 Diafragma y abertura

El diafragma es el dispositivo que permite variar la **abertura**, es decir, permite regular la cantidad de luz que penetrará a través del objetivo hacia la película. La abertura viene expresada en posiciones denominadas números **f** y pueden seleccionarse para controlar la exposición y la profundidad de campo (área nítida de la imagen en torno al objeto enfocado).

La correspondencia entre números **f** y abertura es inversa: cuanto mayor es **f** menor es la abertura y viceversa. Por ello cuando en el lenguaje fotográfico se dice "abrir diafragma" significa aumentar la abertura, es decir, disminuir **f**.

En astrofotografía la abertura es la abertura que ofrece el telescopio.

10.1.3 Obturador

El obturador es el dispositivo que controla el tiempo durante el que llega la luz al elemento sensible. Las cámaras automáticas pueden determinar por sí mismas esa velocidad en función de la abertura del diafragma fijado por el usuario. Al tiempo que el obturador permanece abierto se le llama **tiempo de exposición** y es el tiempo en el que la luz del objeto enfocado estará actuando sobre los productos químicos que contiene la película fotográfica.

A mayor abertura menor tiempo de exposición. Si la velocidad de obturación no es la adecuada la foto quedará demasiado oscura (subexposición, poco tiempo expuesta) o demasiado clara (sobreexposición, demasiado tiempo expuesta).

En relación a la astrofotografía la velocidad de obturación será determinante en el resultado de nuestras fotografías y para ello es aconsejable tener el obturador abierto el mayor tiempo posible.

11 Apéndice 3. Acrónimos

AJAX: Asynchronous JavaScript And XML.

API: Application Programming Interface

CCD : charge coupled device.

CGI: Common Gateway Interface.

DAO: Data Access Object.

DIG: Digital Image Group.

GNU: GNU's NOT Linux.

GWT: Google Web Toolkit.

HDR: High Dinamic Range Imaging.

HFOV: Horizontal Field Of View.

HTML: HyperText Markup Language.

IDE: Integrated Development Environment.

IIP: Internet Image Protocol.

JDBC: Java DataBase Connectivity.

JDK: Java Development Kit.

JNI: Java Native Interface.

JSF: Java Server Faces.

MVC: Model View Controller.

RPC: Remote Procedure Call.

SIFT: Scale-Invariant Feature Transform .

SURF: Speeded-up robust features.

RGB: Red Green Blue.

TIFF: Tagged Image File Format.

ORM: Object Relational Mapping.

QTVR: QuickTime VR

UML: Unified Modelling Language.

WAR: War File.

W3C: Word Wide Web Consortium.
